

**Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Sběr, zpracování dat na platformě Arduino a následné
vykreslení pomocí projektu Grafana**

**Capture, Data Processing on Arduino Platform and following
Graphing with Project Grafana**

2019

Bc. Michaela Kadúchová

Zadání diplomové práce

Student: **Bc. Michaela Kadúchová**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T059 Mobilní technologie

Téma: Sběr, zpracování dat na platformě Arduino a následné vykreslení pomocí projektu Grafana
Capture, Data Processing on Arduino Platform and following Graphing with Project Grafana

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem diplomové práce je zajistit pravidelný sběr a zpracování dat ze sensorů připojených k platformě Arduino. Následně budou data přenášena do počítače, kde budou vykreslována např. pomocí projektu Grafana.

Řešení práce spočívá ve splnění následujících úkolů:

1. Studium a popis problematiky sběru a zpracování dat ze sensorů.
2. Studium a popis problematiky vykreslování grafů.
3. Návrh řešení automatizovaného sběru dat na platformě Arduino.
4. Ověření funkčnosti navrženého řešení a následné zhodnocení.

Seznam doporučené odborné literatury:


- [1] Karvinen T., Karvinen K., Valtokari V. *Make: Sensors: A Hands-On Primer for Monitoring the Real World with Arduino and Raspberry Pi*. Maker Media 2014
- [2] *Grafana Documentation* <http://docs.grafana.org/>

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **Ing. Pavel Nevlud**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2019


prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry




prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlásenie študenta

Prehlasujem, že som túto diplomovú prácu vypracovala samostatne. Uviedla som všetky literárne pramene a publikácie, z ktorých som čerpala.

V Ostrave dňa: 28. apríla 2019

.....*Kováčková*.....
podpis študenta

Pod'akovanie

Rada by som sa poďakovala vedúcemu práce Ing. Pavlovi Nevludovi za odbornú pomoc a konzultáciu pri vytváraní tejto diplomovej práce.

Abstrakt

Táto diplomová práca sa zaoberá zberom a spracovaním dát z vybraných senzorov osvetlenia, teploty, vlhkosti a tlaku. Ďalej návrhom a realizáciou zberu dát z týchto vybraných senzorov. Potom sa zoznámim s aplikáciou mikrokontroléra Arduina UNO ako snímača vybraných fyzikálnych veličín. Práca obsahuje informácie o histórii platformy a technických parametroch. Práca rozoberá I²C komunikáciu, ktorá sa využíva na komunikáciu senzorov s Arduino. Namerané dáta sú ukladané do InfluxDB databázy. Zobrazenie dát prebieha pomocou nástroja Grafana. V závere tejto práce zhodnocujem všetky dosiahnuté výsledky.

Kľúčová slova

Internet vecí, Dáta, Senzor, Arduino, InfluxDB, Grafana

Abstract

This diploma thesis focused on processing of data from selected sensors - luminosity, temperature, humidity and pressure. After, design and implementaion of the data collection system from these sensors. Then get familiar with the application of the Arduino UNO microcontroller as a sensor of physical quantities. Thesis contains information about platform, history and technical parameters. It discussed the I²C communication, which is used for communication sensors with Arduino. The measured data is stored in InfluxDB database. Data visualization is done using Grafana tool. At the end of this thesis I evaluate all the achieved results.

Key words

Internet of Things, Data, Sensor, Arduino, InfluxDB, Grafana

Obsah

Zoznam použitých skratiek a symbolov	- 8 -
Zoznam obrázkov.....	- 10 -
Zoznam tabuliek a zoznam výpisov	- 12 -
Úvod	- 13 -
1 Zber a spracovanie dát zo senzorov	- 14 -
1.1 Dáta zo senzorov	- 14 -
1.2 Databázy časových radov	- 15 -
1.3 Výber databázového systému	- 19 -
1.4 Použité senzory.....	- 20 -
2 Vývojová platforma Arduino	- 23 -
2.1 Vývojové dosky Arduino.....	- 23 -
2.2 Shieldy.....	- 31 -
2.3 Arduino IDE	- 33 -
2.4 Komunikačné rozhranie	- 34 -
3 Vizualizačný nástroj Grafana	- 37 -
3.1 Inštalácia a konfigurácia	- 37 -
3.2 Data Sources.....	- 38 -
3.3 Graph Panel	- 40 -
3.4 Dashboard.....	- 41 -
4 Návrh riešenia automatizovaného zberu dát	- 42 -
4.1 Realizácia modulu	- 42 -
4.2 Návrh riešenia.....	- 43 -
4.3 Zapojenie senzorovej časti.....	- 44 -
4.4 Programové vybavenie	- 46 -
4.5 Zber a uloženie dát	- 49 -
4.6 Zobrazovanie dát	- 52 -
Záver	- 57 -
Použitá literatúra	- 58 -
Zoznam príloh	- 61 -

Zoznam použitých skratiek a symbolov

Skratka	Význam
API	Applicatoin Programming Interface
C	Programovací jazyk
C++	Multiparadigmatický programovací jazyk
CCD	Charge-coupled device
CLI	Command Line Interface (Rozhranie príkazového riadku)
COM	Communication port (Komunikačný port)
DBMS	Database Management System
EDBP	Atmel Embedded Debbuger
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
GUI	Graphical User Interface (Grafické užívateľské rozhranie)
HID	Human interface device
ICSP	In-Circuit Seriál Programming
I/O pin	Input/Output pin (Vstupne/Výstupný pin)
PC	Inter-Integrated Circuit (Multi-masterová počítačová sériová zbernica)
IoT	Internet of Thing
IDE	Integrated Development Environment (Vývojové prostredie)
LCP	Liquid crystal display
MISO	Vstup riadiaceho a výstup riadeného (Master In, Slave Out)
MOSI	Výstup riadiaceho a vstup riadeného (Master Out, Slave In)
NoSQL	Not Only SQL
RAM	Random Access Memory (Pamäť s náhodným prístupom)
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory (Statická pamäť)
SS, CS	Kontrolná linka pre riadenia (slave select), (chip select)
TSD	Time Series Daemon

Zoznam použitých skratiek a symbolov

TSDB	Time Series Database
UART	Universal Asynchronous Receiver and Transmitter (Univerzálny asynchrónny prijímač/vysielač)
USB	Universal Serial Bus
VM	Virtual Machine (virtuálny stroj)
Wi-Fi	Wireless Fidelity (Bezdrátová počítačová sieť)

Zoznam obrázkov

Číslo obrázku	Názov obrázku
1.1	Graf zobrazujúci rýchlosť zápisu (prevzatý z [3])
1.2	Graf zobrazujúci nároky na diskový priestor (prevzatý z [3])
1.3	Webové rozhranie databázy InfluxDB
1.4	Senzor TSL 2561 (prevzatý z [13])
1.5	Schéma zapojenia senzoru TSL 2561
1.6	Spektrálna analýza fotodiód použitých v snímači TSL2561
1.7	Bloková schéma snímača intenzity osvetlenia TSL2561
1.8	Senzor BME 280 (prevzatý z [15])
1.9	Schéma zapojenia senzoru BME 280
2.1	Arduino Mini (prevzatý z [20])
2.2	Arduino Nano (prevzatý z [22])
2.3	Arduino Micro (prevzatý z [23])
2.4	Arduino LilyPad (prevzatý z [23])
2.5	Arduino Fio (prevzatý z [23])
2.6	Arduino UNO
2.7	Arduino Leonardo (prevzatý z [20])
2.8	Arduino Yún (prevzatý z [20])
2.9	Arduino Zero (prevzatý z [22])
2.10	Arduino Mega256 (prevzatý z [22])
2.11	Arduino Due (prevzatý z [22])
2.12	Arduino Esplora (prevzatý z [20])
2.13	Arduino Robot (prevzatý z [20])
2.14	Arduino Intel Galileo (prevzatý z [20])
2.15	WiFi Shield (prevzatý z [24])
2.16	Motor Shield (prevzatý z [25])
2.17	Ethernet Shield (prevzatý z [26])
2.18	Relé Shield (prevzatý z [21])

2.19	GSM Shield (prevzatý z [21])
2.20	Arduino IDE
2.21	Komunikačné zbernice (prevzatý z [27])
2.22	SPI komunikácia (prevzatý z [28])
2.23	I ² C komunikácia
3.1	Grafana - prihlásenie
3.2	Grafana - zdroj dát
3.3	Grafana - Graph Panel
3.4	Grafana - ukázkový dashboard
4.1	Zostava základnej dosky Arduino UNO a použitých senzorov
4.2	Model monitorovacieho nasadenia
4.3	Bloková schéma
4.4	Grafická schéma pripojenia s Arduino
4.5	Elektrická schéma zapojenia senzoru TSL 2561
4.6	Elektrická schéma zapojenia senzoru BME 280
4.7	Vývojový diagram funkčnosti aplikácie
4.8	Výpis debug hlášky z terminálu
4.9	GUI - ukážka prvých desiatich hodnôt v InfluxDB
4.10	CLI - ukážka prvých desiatich hodnôt v InfluxDB
4.11	Ukážka agregácie hodnôt (podľa priemeru)
4.12	Okno vizualizácie - Temperature
4.13	Okno vizualizácie - Humidity
4.14	Okno vizualizácie - Pressure
4.15	Okno vizualizácie - Luminosity
4.16	Ukážka reportu teploty
4.17	Ukážka reportu vlhkosti
4.18	Ukážka reportu tlaku
4.19	Ukážka reportu svetivosti

Zoznam tabuliek a zoznam výpisov

Číslo tabuľky	Názov tabuľky
A.1	Najpoužívanejšie databázy časových radov (prevzaté z [3])
A.2	Porovnávací prehľad dosiek Arduina
A.3	Piny Arduina pre SPI komunikáciu
A.4	Piny Arduina pre I2C komunikáciu

Číslo výpisu	Názov výpisu
B.1	Konfigurácia úložiska InfluxData
B.2	Inštalácia InfluxDB
B.3	Vytvorenie databázy
B.4	Inštalácia Grafany
B.5	Potrebné knižnice pre použité senzory
B.6	Programové vybavenie - cyklus setup()
B.7	Programové vybavenie - cyklus loop()
B.8	Programové vybavenie - pseudokód
B.9	Implementácia InfluxDBSyncer - pseudokód

Úvod

Pojem Internet vecí (Internet of Things) je v dnešnej dobe veľmi používaný pojem v oblasti informatiky. Pre tento termín neexistuje žiadna ustálená definícia, ale dá sa pochopiť ako sieť fyzických zariadení so zabudovanou internetovou konektivitou, ktoré sú opatrené senzormi a softwarom, ktorý umožňuje ich vzájomnú komunikáciu. Princípom je zber dát z rôznych senzorov a čidiel a zdieľanie týchto dát prostredníctvom internetu za účelom ďalšieho spracovania a vyhodnocovania. Základom celého IoT sú dáta. Dáta zo senzorov sú neoddeliteľnou súčasťou rastúcej reality prostredia internetu vecí.

Cieľom diplomovej práce je zaistiť pravidelný zber a spracovanie dát zo senzorov. Zber dát je realizovaný na platforme Arduino. Dáta sú následne prenášané do počítača, kde sú vykresľované pomocou projektu Grafana.

Práca sa zaoberá rozborom funkcie digitálnych senzorov, ktoré sa najčastejšie používajú v sieťach IoT. Dnešný trh ponúka nespočetné množstvo senzorov. Pre prácu je zvolený svetelný senzor TSL 2561 a senzor BME 280, ktorý zahŕňa 3 typy senzorov, a to teplotný, tlakový a vlhkostný senzor.

Arduino je mikrokontrolér, ktorý bude slúžiť k zberu dát. Arduino je otvorená vývojová platforma, založená na užívateľsky nenáročnom hardwaru a softwaru. Dosky Arduino sa objavili ako lacná cvičná doska, ktorú môžu študenti a nenároční užívatelia používať pre svoje experimenty. Rozsah dnešnej ponuky dosiek Arduino je veľký a dosky je možné použiť v širokej oblasti aplikácií.

Diagnostickým nástrojom je monitorovací systém, ktorý bude zobrazovať hodnoty v rámci dlhšieho časového úseku a súčasne ich bude archivovať do databázy. Načítanie dát do databázy bude zaisťovať skript, napísaný v jazyku Python. Dáta sú uchovávané v NoSQL databázy InfluxDB. Vizualizovať dáta z databázy je možné pomocou nástroja Grafana. Webová aplikácia pracuje so všetkými získanými dátami z mikropočítača, medzi ktoré patrí teplota, vlhkosť, tlak a svetelná intenzita.

Posledná kapitola bude venovaná samotnému návrhu zariadenia, spôsobu komunikácie medzi blokmi zariadenia a princípom fungovania jednotlivých častí.

1 Zber a spracovanie dát zo senzorov

Senzor je časť zariadenia, ktorá poskytuje informácie o podnetoch z vonkajšieho prostredia a informácie o vlastnostiach tohto prostredia.

1.1 Dáta zo senzorov

Dáta zo senzorov sú výstupom zariadenia, ktoré detekuje a reaguje na určitý typ vstupu z fyzického prostredia. Výstup môže byť použitý na poskytnutie informácií alebo vstupov do iného systému alebo na vedenie procesu.

Senzory môžu byť použité na detekciu akéhokoľvek fyzického prvku.

Akcelerometer sníma zmeny gravitačného zrýchlenia v zariadení, ktorý je nainštalovaný napríklad v smartphone alebo v hernom ovládači, k zisteniu zrýchlenia, sklonu a vibrácií. V prvých zariadeniach nachádzal minimálne uplatnenie, postupom času sa využíva na automatické preklápanie obrazovky, hranie hier a využívajú ho aplikácie ako krokomer a pod.

Fotosenzor detekuje prítomnosť viditeľného svetla, infračerveného prenosu alebo ultrafialovej energie.

Lidar je laserová metóda detekcie, detekcia rozsahu a mapovanie, zvyčajne používa laser s nízkym výkonom a bezpečným pre oko v spojení s kamerou.

Zariadenie viazané na náboj (CCD) ukladá a zobrazuje dáta pre obraz takým spôsobom, že každý pixel sa prevedie na elektrický náboj, ktorého intenzita súvisí s farbou vo farebnom spektre.

Snímače inteligentnej siete môžu poskytovať údaje v reálnom čase o podmienkach siete, zisťovanie výpadkov, poruchy a zaťaženie a spúšťanie alarmov.

Bezdrôtové senzorové siete kombinujú špecializované snímače s komunikačnou infraštruktúrou pre monitorovanie a zaznamenávanie podmienok na rôznych miestach. Bežne sledované parametre zahŕňajú teplotu, vlhkosť, tlak, smer a rýchlosť vetra, intenzitu osvetlenia, intenzitu vibrácií, intenzitu zvuku, napätie elektrického vedenia, chemické koncentrácie, hladiny znečisťujúcich látok a životne dôležité funkcie tela. [1]

Senzorické informácie sú typicky kategórie „time series“, jednoduché správy obsahujúce nejakú nameranú hodnotu. Každý senzor musí mať identifikáciu a kategorizáciu, teda akúkoľvek informáciu, podľa ktorej potom chceme triediť a vyhľadávať. Druhý typ informácie je čo meriame, teda teplotu, vlhkosť, obsah oxidu uhličitého apod. Tretie informácia je samotná nameraná hodnota.

V prípade „time series“ dát je základnou dimenziou vždy čas. Pri time series dátach prakticky nikdy nie je potrebné modifikovať nejaký záznam ani selektívne mazať. Data zo senzorov ostávajú rovnaké, odmazávajú sa veľké bloky zastaralých dát. V každom prípade sa očakáva veľké množstvo stále rastúcich dát so zväčšujúcou sa časovou značkou, takže systém by mal

disponovať veľkou škálovateľnosťou a automatickým premazávaním. Napríklad udržiavať údaje za posledný týždeň. [2]

1.2 Databázy časových radov

Pre „time series“ sú špecifické operácie, na ktoré tradičné databázy nie sú optimalizované. Za účelom monitorovania sa používajú špeciálne časozberné databázy postavené na modeli kľúč-hodnota. Každý takejto dvojici je ďalej možné priradiť radu štítkov. Tieto databázy sú prispôbené pre dlhodobý zber dát s časovými údajmi a poskytujú funkcionality pre prácu s týmito dátami. Tá zahŕňa mimo dotazovania tiež agregáciu a zahadzovanie starých dát. [2]

Time Series databázy sú najrýchlejšie rastúcim segmentom databázového segmentu. Existuje mnoho spôsobov ako určiť, ktorá databáza časových radov je najlepšia. DB-Engines radia databázy založené na popularite vyhľadávani, zmien v sociálnych médiách alebo pracovných pozíciách. Poradie býva aktualizované mesačne. Aktuálne výsledky sú zobrazené v tabuľke A.1.

Tabuľka A.1: *Najpoužívanéjšie databázy časových radov (prevzaté z [3])*

poradie	DBMS	výsledok	
		marec 2019	24 mesiacov
1	InfluxDB	16,17	+9,00
2	Kdb+	5,60	+4,03
3	Graphite	3,07	+1,21
4	RRDtool	2,75	+0,02
5	Prometheus	2,72	+2,12
6	OpenTSDB	2,28	+0,72
7	Druid	1,57	+0,73
8	TimeScaleDB	0,91	+0,88
9	KairosDB	0,66	+0,24
10	FaunaDB	0,52	+0,28

OpenTSDB databázy

OpenTSDB, alebo skratka pre open time series database, je databáza postavená nad distribuovanou nerelačnou databázou HBase. OpenTSDB nie je samostatná databáza časových radov. Jedná sa o akýsi druh nadstavby, ktorý sa zameriava čisto na časovo usporiadané dáta. Na serveroch, z ktorých chce získavať údaje, bežia jeho klienti, ktorí v intervaloch posielajú namerané hodnoty zberným démonom (TSD) a tí sa už starajú o ukladanie dát v NoSQL databáze HBase. Tento systém je výhodný hlavne svojou škálovateľnosťou, je možné nasadiť toľko TSD, koľko môže byť poskytnutých serverov. OpenTSDB obsahuje vlastné webové rozhranie, ktoré umožňuje základné vykresľovanie grafov z uložených dát. [4]

NoSQL databázy

Pod termínom NoSQL viacero zdrojov uvádza, že tento pojem vychádza zo slov "noSQL" (nie-SQL databázy) alebo "not only SQL" (nielen SQL). NoSQL databázy boli vyvinuté tak, aby vyhovovali novým požiadavkám, vychádzajúcich z tvorby moderných webových aplikácií. Medzi základné charakteristiky, ktoré sú spoločné pre väčšinu NoSQL databáz:

- sú open source,
- nepoužívajú relačný model (navyše ani SQL),
- sú navrhnuté na prácu v klastri,
- umožňujú vloženie dát bez vopred preddefinovanej schémy (záznamy nie sú obmedzené presnou štruktúrou),
- sú horizontálne škálovateľné, čo znamená, že databáza automaticky ukladá jednotlivé fragmenty na rôzne servery.

Ďalej podpora automatickej replikácie dát (duplikovanie serverov), jednoduché programovacie rozhranie (API) a schopnosť spracovávať veľké objemy dát patria medzi ďalšie vlastnosti NoSQL databáz. [5]

Pre ukladanie informácií, ktoré zariadenia zaznamenali v nejakom časovom horizonte, je vhodné použiť databázu zo skupiny NoSQL. Klasické SQL databázy rastú smerom nadol, pridávaním riadkov. Najznámejšie NoSQL databázy sú MongoDB a InfluxDB. [6]

MongoDB

MongoDB je open-source dokumentovo orientovaná databáza vyvinutá spoločnosťou 10gen, jedna z najpoužívanejších NoSQL databáz. MongoDB je univerzálny sklad dokumentov. MongoDB má veľkú používateľskú základňu a disponuje vlastnosťami NoSQL databáz, ktoré ju robia vhodnou voľbou pre registračný portál, ktorý potrebuje rýchlu, škálovateľnú databázu s určitou redundanciou na zvýšenie odolnosti dát. [7]

InfluxDB

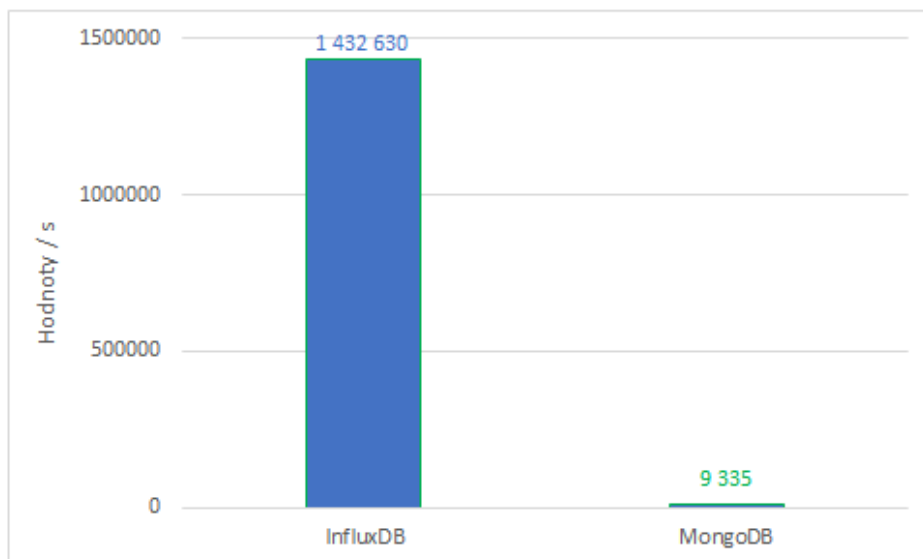
InfluxDB je open-source databáza použitá v systéme spoločnosti Influx. Napriek tomu, že samotná databáza je open-source, funkcionality, umožňujúca systém distribuovať a horizontálne škálovať, je podporovaná len v platenej verzii. Databáza je napísaná v jazyku Go, optimalizovaná na rýchle a efektívne pracovanie s údajmi v čase, ako napr. údaje z monitorovania, analýz či zbieraní dát zo senzorov v IoT. Je postavená špeciálne na ukladanie údajov z časových radov. Medzi hlavné výhody InfluxDB patrí:

- rýchla inštalácia (nemá žiadne závislosti na iné balíčky),
- pre prácu s ňou poskytuje jazyk podobný SQL,
- podpora spracovanie 1 milióna hodnôt za sekundu.

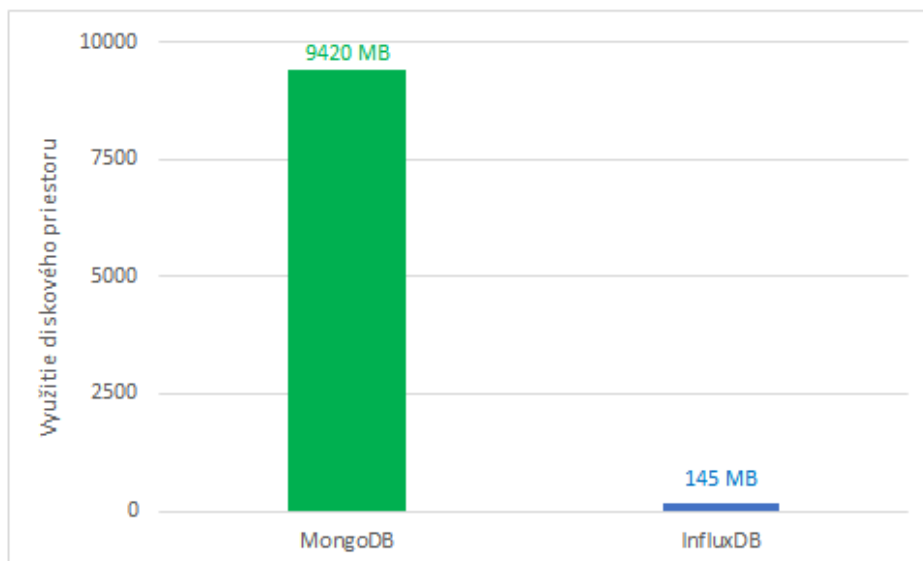
Klasické NoSQL databázy, ako napr. MongoDB, by bolo potrebné na prácu s hodnotami v čase najprv nakonfigurovať. Konfigurácia vopred vyžaduje rozhodnutia o štruktúrovaní kolekcii a datových typoch, čo môže byť časovo náročné. Navyše by to malo dlhodobé vplyvy na to ako by sa s databázou muselo pracovať. Na druhej strane, InfluxDB je po inštalácii schopná pracovať

ihneď. Podľa záťažových testov vykonaných v priebehu 6 hodín na počte 100 serverov, pričom každý server meral 100 rôznych veličín a posielal dáta do databázy každých 10 sekúnd má InfluxDB v porovnaní s MongoDB:

- 2,4 – krát rýchlejší zápis (Obr. 1.1),
- 20 – krát lepšiu kompresiu dát (Obr. 1.2),
- 5,7 – krát lepší výkon čo sa týka rýchlosti spracovania dotazov. [6]



Obrázek 1.1: Graf zobrazujúci rýchlosť zápisu (prevzatý z [3])



Obrázek 1.2: Graf zobrazujúci nároky na diskový priestor (prevzatý z [3])

Údaje ukázali, že InfluxDB výrazne prekonala MongoDB pri prijímaní údajov a ukladaní na disku. Na ukladanie hodnôt v čase a ich spracovávanie je InfluxDB momentálne najvhodnejšia voľba.

Relačné databázy

Oproti predchádzajúcim spomenutým sú tieto riešenia založené na skutočných databázových systémoch, a teda vo všeobecnosti poskytujú oveľa silnejšie garancie persistentnosti a efektívnosti. Databáza obsahuje dimenzionálnu tabuľku obsahujúcu identifikátory a detaily o spracúvaných časových radoch a inú, tzv. tabuľku faktov, obsahujúcu samotné záznamy spolu s identifikátormi časových radov, ku ktorým patria.

Databázové systémy používajúce relačný model a SQL rozhranie garantujúce ACID vlastnosti transakcií, nie sú príliš všeobecné a mnoho z funkcií, ktoré poskytujú, systém pre prácu s časovými radmi nevyužije. Taktiež majú vo všeobecnosti problém s horizontálnym škálovaním.

Tgres

Tgres je open-source rozšírenie databázového systému PostgreSQL, určené na uchovávanie časových radov, ktoré je momentálne vo vývoji. Vizualizácia uložených dát je zabezpečená pomocou prostriedku Grafana. [8]

Wire-column NoSQL databáza

Cassandra je column databáza založená na kľúč-hodnota. Riešenie Cassandra je určené pre spravovanie veľkých datových objemov vytážených cez viac uzlov bez jediného bodu zlyhania. Jednou z najväčších predností Cassandry je schopnosť zvládnuť obrovské množstvo neštruktúrovaných dát. Senzor sa uloží na jeden riadok a každé meranie bude nový stĺpec, čo bude škálovať až do milión stĺpcov. [2] [9]

NewSQL databázy

New SQL tvoria skupinu nových vysoko škálovateľných transakčných databázových systémov. Podstatou NewSQL je podpora ACID transakcií a možného horizontálneho škálovania. Ich hlavným využitím sú systémy, kde je vyžadované veľké množstvo transakcií, ale ich komplexnosť nie je príliš vysoká.

Vertica

Vertica je zaujímavou variantom NewSQL databázy. Tá je v princípe relačná a má ACID vlastnosti, takže sa môže použiť i tam kde je zvláštny dôraz na konzistenciu časozberných dát, napr. pre forenzné účely. Na rozdiel od klasických SQL systémov je ale databáza optimalizovaná pre analytické operácie (napr. priemerné teploty za určité obdobie). Vertica má špeciálny režim pre time series operácie a dokáže dokonca sama lineárne interpolovať chýbajúce údaje. [2]

1.3 Výber databázového systému

InfluxDB je time series databáza v najnovšej verzii postavená nad BoltDB. Jej prednosťou oproti ostatným databázovým systémom je jednoduchý dotazovací jazyk, ktorý vychádza zo syntaxe SQL. Skupiny bodov sa zhromažďujú v meraniach a existujú funkcie, ktoré z meraní vytvárajú nové s hrubšou granularitou. Tagy sú implementované až v najnovšej verzii 0.9. Je zrejmé, že najlepšou voľbou je InfluxDB, i keď je ešte vo vývoji. [4]

Inštalácia a konfigurácia systému InfluxDB

Inštalácia InfluxDB je podrobne zdokumentovaná na oficiálnej stránke [10]. Vo výpise B.1 sú potrebné príkazy pre konfiguráciu úložiska InfluxData. Vo výpise B.2 sú príkazy pre inštaláciu a spustenie služby InfluxDB.

Výpis B.1: *Konfigurácia úložiska InfluxData*

```
$ curl -sL https://repos.influxdata.com/influxdb.key
$ sudo apt-key add -
$ source /etc/lsb-release
$ echo "deb
https://repos.influxdata.com/${DISTRIB_ID,,}${DISTRIB_CODENAME}
stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
```

Výpis B.2: *Inštalácia InfluxDB*

```
$ sudo apt-get update && sudo apt-get install influxd
$ sudo systemctl start influxdb
```

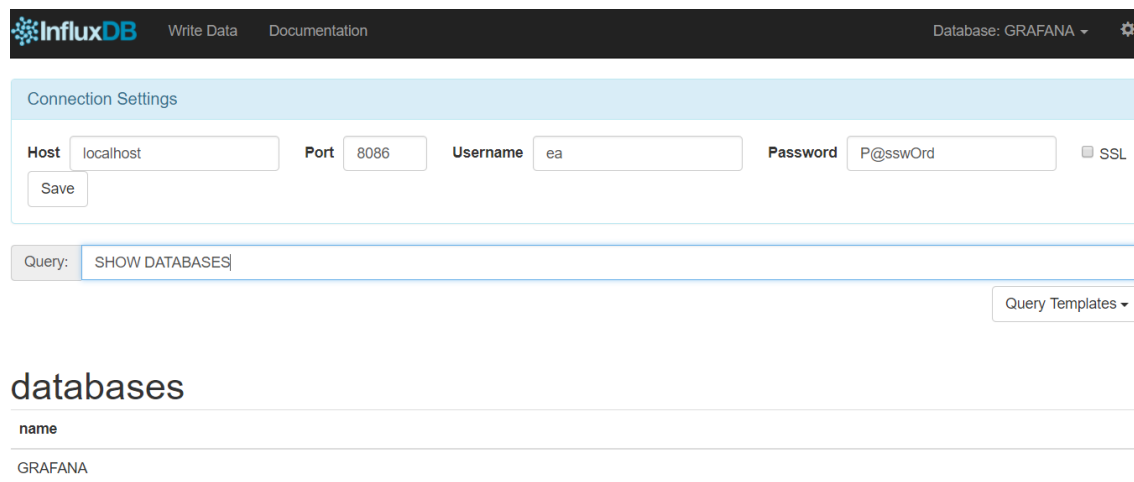
InfluxDB bola nainštalovaná na webový server Apache. InfluxDB má pre prvé spustenie otvorené porty 8083 (GUI) a 8086 (API). Po inštalácii je aplikácia spustená s predvolenými nastaveniami a pripravená pracovať. Prvými krokmi sú vytvorenie databázy. Vo výpise B.3 je ukážka vytvorenia databázy.

Výpis B.3: *Vytvorenie databázy*

```
> CREATE USER "ea" WITH PASSWORD 'P@sswOrd' WITH ALL PRIVILEGES
> auth ea P@sswOrd
> create database GRAFANA
```

InfluxDB pred spustením nie je potrebné nijak zvlášť nastavovať. Bolo ale potrebné previesť nastavenie databázy, čo je jednoducho prevediteľné cez webové GUI, ktoré je dostupné na adrese: *http://localhost:8083*.

Ukážka webového rozhrania databázy InfluxDB je na obr. 1.3. V konfiguračnom okne sa nastavil *Host*, *Port*, *Username* a *Password*.



The screenshot shows the InfluxDB web interface. At the top, there's a navigation bar with 'InfluxDB', 'Write Data', and 'Documentation' links. On the right, it says 'Database: GRAFANA' with a settings icon. Below this is a 'Connection Settings' section with input fields for 'Host' (localhost), 'Port' (8086), 'Username' (ea), and 'Password' (P@sswOrd). There's an 'SSL' checkbox and a 'Save' button. Below the settings is a 'Query' section with a text input containing 'SHOW DATABASES' and a 'Query Templates' dropdown. The query result is displayed as a table with one row: 'name' with the value 'GRAFANA'.

name
GRAFANA

Obrázek 1.3: Webové rozhranie databázy InfluxDB

1.4 Použité senzory

Pre zber dát boli vybrané dva senzory. TSL 2561 je svetelný senzor použiteľný takmer vo všetkých situáciách. V porovnaní s nízkonákladovými CdS, je ďaleko presnejší. Ďalší použitý senzor BME 280 je vynikajúci a cenovo dostupný senzor umožňujúci okrem snímania teploty, vlhkosti vzduchu i meranie barometrického tlaku. [11]

Senzor meranie osvetlenia TSL 2561

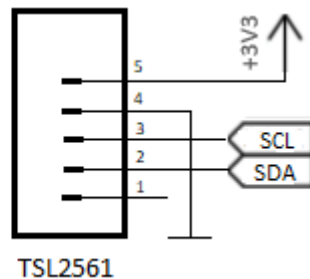
Intenzita osvetlenia je snímaná senzorom TSL2561 od firmy TAOS. Ide o pokročilý snímač osvetlenia, vďaka nastaviteľnému zosilneniu a premennej integračnej dobe môže merať intenzitu osvetlenia v rozsahu 0,1 až 40 000 *Lux*. [12]



Obrázek 1.4: Senzor TSL 2561 (prevzatý z [13])

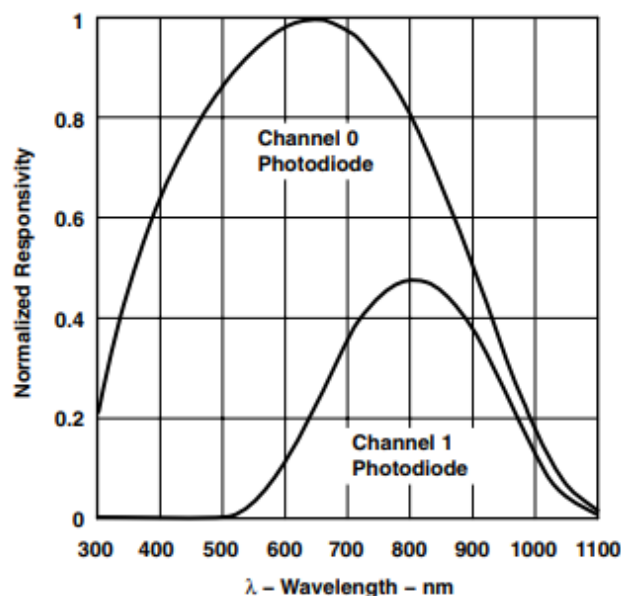
Vlastnosti TSL 2561 senzora:

- blíži sa vlastnostiam ľudského oka,
- presne meria osvetlenie v rôznych svetelných podmienkach,
- teplotný rozsah: $-30\text{ }^{\circ}\text{C}$ až $80\text{ }^{\circ}\text{C}$,
- dynamický rozsah: 0,1 až 40 000 Lux,
- napäťový rozsah: 2,7 – 5 V,
- rozhranie: I²C,
- použitá I²C 7-bitové adresy 0x39, 0x29, 0x49, voliteľné pomocou jumperov. [11]

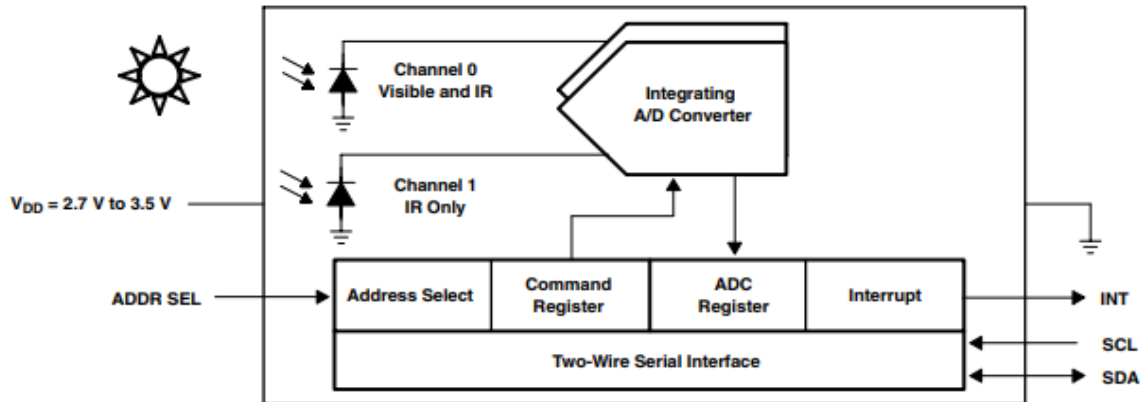


Obrázek 1.5: Schéma zapojenia senzoru TSL 2561

Obsahuje dve snímacie diódy, jednu infračervenú a druhú širokospektrálnu. Môže sa merať oddelene infračervené, celospektrálne alebo viditeľné svetlo. Väčšina senzorov môže snímať iba jedno zo svetiel, čo neodpovedá ľudskému oku. Na Obr. 1.6 je znázornená spektrálna charakteristika snímača intenzity osvetlenia TSL2561 z oboch fotodiód. Funkčná bloková schéma snímača intenzity osvetlenia TSL2561 je uvedené na Obr. 1.7. [12]



Obrázek 1.6: Spektrálna analýza fotodiód použitých v snímači TSL2561 (prevzatý z [14])



Obrázek 1.7: Bloková schéma snímača intenzity osvetlenia TSL2561 (prevzatý z [14])

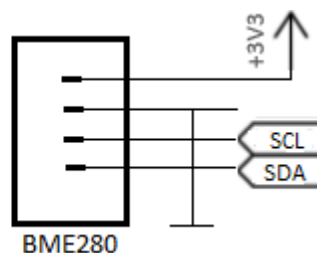
Senzor teploty, vlhkosti a tlaku BME 280

Integrovaný senzor BME 280 od spoločnosti Bosch v sebe zahŕňa 3 typy senzorov: teplotný, tlakový, vlhkosťový senzor. Tento merací senzor komunikuje cez rozhranie I²C a zvláda komunikovať rýchlosťou až 3,4 MHz.



Obrázok 1.8: Senzor BME 280 (prevzatý z [15])

Čo sa týka meracích a prevádzkových rozsahov senzoru BME280, tak u teploty je to -40 až $+85$ stupňov Celzia, u vlhkosti 0 až 100%, a tlak sa môže merať v rozsahu 300 až 1100 hektoPascalov. Rozlíšenie u meraných veličín je 0,01 stupňa Celzia u teploty, 0,008% u vlhkosti a 0,18 Pacalov u tlaku. Presnosť týchto veličín je potom ± 1 stupeň Celzia u teploty, $\pm 3\%$ u vlhkosti a ± 1 Pascalov u tlaku. [15]



Obrázok 1.9: Schéma zapojenia senzoru BME 280

2 Vývojová platforma Arduino

Arduino je vývojová platforma, ktorá vznikla v roku 2005 v Taliansku v meste Ivrea. Zakladatelia Massimo Banzi, David Cuartielles pomenovali projekt po Arduinovi Ivrejskom, významnej historickej postave mesta. Cieľom projektu bolo vytvoriť jednoduchú platformu pre študentov, ktorá umožní rýchly vývoj a jednoduché používanie. Medzi študentami sa Arduino uchytilo, a tak sa tvorcovia rozhodli poskytnúť ho celému svetu. Projekt je už od začiatku voľne dostupný (open-source), dokumentácia a referenčná príručka jazyka a externých knižníc je potom väčšinou vydávaná pod licenciou Creative Commons. V roku 2010 vyšla jedna z prelomových dosiek, ktorá mala označenie UNO a je, dá sa povedať jednou z najpoužívanejších. [16] [17]

Výhody Arduina:

- jednoduché a zrozumiteľné vývojové prostredie,
- široká podpora komunity,
- relatívne nízka cena,
- open-source projekt,
- USB pripojenie,
- rýchly vývoj a dostupnosť modulov (shieldov) v podobe snímačov, konvertorov medzi komunikačnými protokolmi,
- existencia niekoľkých emulátorov a simulátorov, nahradzujúcich časť funkcionality reálnej dosky Arduina (napr. Arduino IDE, CodeBlocks).

Arduino má i radu nevýhod:

- projekty Arduino nedisponujú príliš dlhou životnosťou,
- Arduino nie je vhodné pre komerčné aplikácie, pretože tie väčšinou vyžadujú certifikáciu k splneniu požiadaviek,
- v súčasnosti platforma neponúka možnosť použitia šifrovacích protokolov,
- V posledných rokoch sa na trh dostávajú nekvalitné produkty, vyrábané prevažne v Číne. [18]

2.1 Vývojové dosky Arduino

Väčšina dosiek Arduino má modrú až modro-zelenú farbu a sú veľmi často označené prídavným názvom *Rev3* popr. *R3*. Toto označenie znamená, že ide o tú istú verziu, ale v novšom vydaní. Jedná sa o malú zmenu, vydanie novej dosky je zbytočné. Medzi jednotlivými verziami sa mohol napríklad zmeniť design, alebo rozloženie súčiastok. Funkčnosť dosiek ostane rovnaká. Domovská webová stránka projektu Arduino je www.arduino.cc.

Dosky Arduino obsahujú 8 – bitové mikrokontroléry z rodiny AVR od firmy Atmel a množstvo ďalších podporných obvodov. Oficiálne vydanie Arduino, ktoré vyrába a predáva Talianska firma Smart Projects, používajú čipy ATmega8, ATmega168, ATmega328, ATmega1280 a ATmega2560. Z mikroprocesoru je väčšina I/O pinov vedená na konektory

pre pripojenie ďalších periférií, ktorým sa vo svete Arduino hovorí tzv. shieldy. Ide o moduly, ktoré sa nasúvajú na dosku Arduina a obsahujú rozširujúce periférie napr. WiFi, LCD display. Z podporných obvodov na doske býva niekoľko diód, resetovacie tlačidlo, konektory pre ICSP programovanie, napájací konektor, oscilátor a obvod zaistujúci komunikáciu po USB. [19]

Nasledujúca tabuľka A.2 znázorňuje porovnanie dosiek Arduino v niekoľkých parametroch.

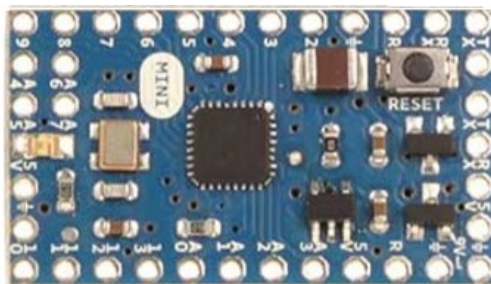
Tabuľka A.2: *Porovnávací prehľad dosiek Arduina*

Typ dosky	Mikrokontrolér	FLASH [kB]	SRAM [kB]	I/O piny		Rozmery [mm]
				digitálne	analogové	
Mini	ATmega328	32	2	14	8	30 × 18
Nano	Atmega328P	32	2	14	8	45 × 18
Micro	ATmega32u4	32	2,5	20	12	48 × 18
LilyPad	ATmega328V	16	1	14	6	–
Fio	Atmega328P	32	2	14	8	28 × 65
UNO	Atmega328P	32	2	14	6	68 × 53
Leonardo	ATmega32u4	32	2,5	20	12	68 × 53
Yún	ATmega32u4	32	2,5	20	12	68 × 53
Zero	ATSAMD21G18A	256	32	14	6	68 × 53
Mega2560	Atmega2560	256	8	54	16	101 × 53
Due	AT91SAM3X8E	512	96	54	12	101 × 53
Esplora	ATmega32u4	32	2,5	–	–	164 × 60

Dôležitosť dosiek sa môže posudzovať z mnoho hľadísk ako napríklad rozšírenosť medzi užívateľmi, najlepšia podpora komunity, najkvalitnejšie komponenty, alebo najväčší potenciál. Pre diplomovú prácu som vybrala Arduino UNO.

Arduino Mini

Arduino Mini je najmenšia oficiálna verzia Arduina a je navrhnutá pre úsporu miesta. Doska bola pôvodne založená na ATmega168, teraz beží na procesore ATmega328 s taktom 16 MHz. Pre malé rozmery nemá na doske USB port, k programovaniu je nutné použiť USB 2 Seriál prevodník. Táto doska je určená k použitiu v chytrých vypínačoch, diaľkových ovládačoch a všeobecne k všetkých zariadení, kde je potrebná čo najmenšia doska. [20]



Obrázok 2.1: *Arduino Mini (prevzatý z [20])*

Arduino Nano

Arduino Nano má na rozdiel od Arduina Mini USB port a prevodník, pomocou ktorého sa dá pripojiť k počítaču cez USB. Je to komfortnejšie, pretože sa už nemusí používať externý prevodník pre programovanie. Rýchlosť dosky je rovnaká ako u predchádzajúcej. Arduino Nano má podobné funkcie ako Arduino UNO. Chýba len jednosmerný napájací konektor a pracuje s káblom Mini-B USB namiesto štandardného. Jedná sa o malú, ale kompletnú Arduino dosku. [21]



Obrázok 2.2: *Arduino Nano (prevzatý z [22])*

Arduino Micro

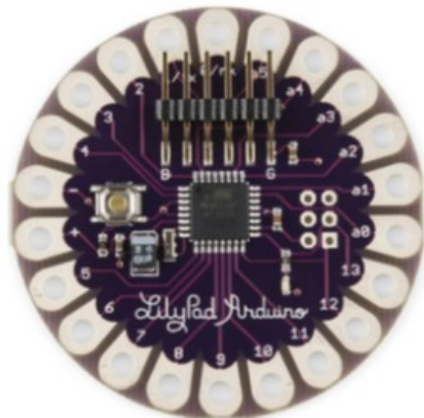
Arduino Micro bolo vytvorené v spolupráci s Adafruit. Je jedna z dosiek, ktorá má čip obsahujúci prevodník. Týmto čipom je ATmega32u4, vďaka ktorému je Micro rozpoznateľný ako klávesnica alebo myš. Micro doska je podobná Arduino Leonardo v tom, že ATmega32U4 má vstavanú USB komunikáciu, čo eliminuje potrebu sekundárneho procesora. S touto doskou je veľmi jednoduché vytvoriť si vlastnú klávesnicu, alebo herný počítač. [21]



Obrázok 2.3: *Arduino Micro (prevzatý z [23])*

LilyPad Arduino

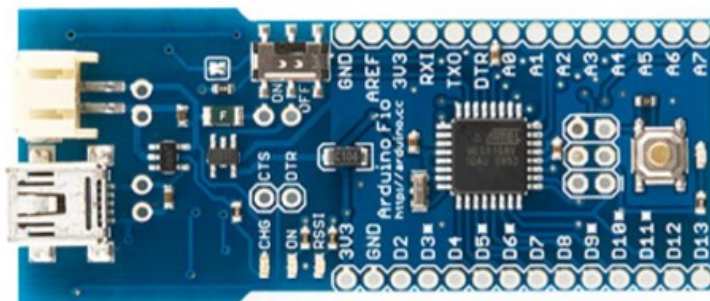
LilyPad Arduino nie je tak úplne typická doska. Bola navrhnutá pre nositeľné aplikácie. Spoje na tejto doske sú tvorené vodivou niťou. Pracuje na nabíjateľných batériách a umožňuje jednoduché pripojenie k snímačom a ovládačom vyvinutým pre jednoduchú integráciu do oblečenia a textílií. Tak sa dá vyrobiť napríklad cyklistická mikina s prišitými smerovkami. [23]



Obrázok 2.4: *Arduino LilyPad (prevzatý z [23])*

Arduino Fio

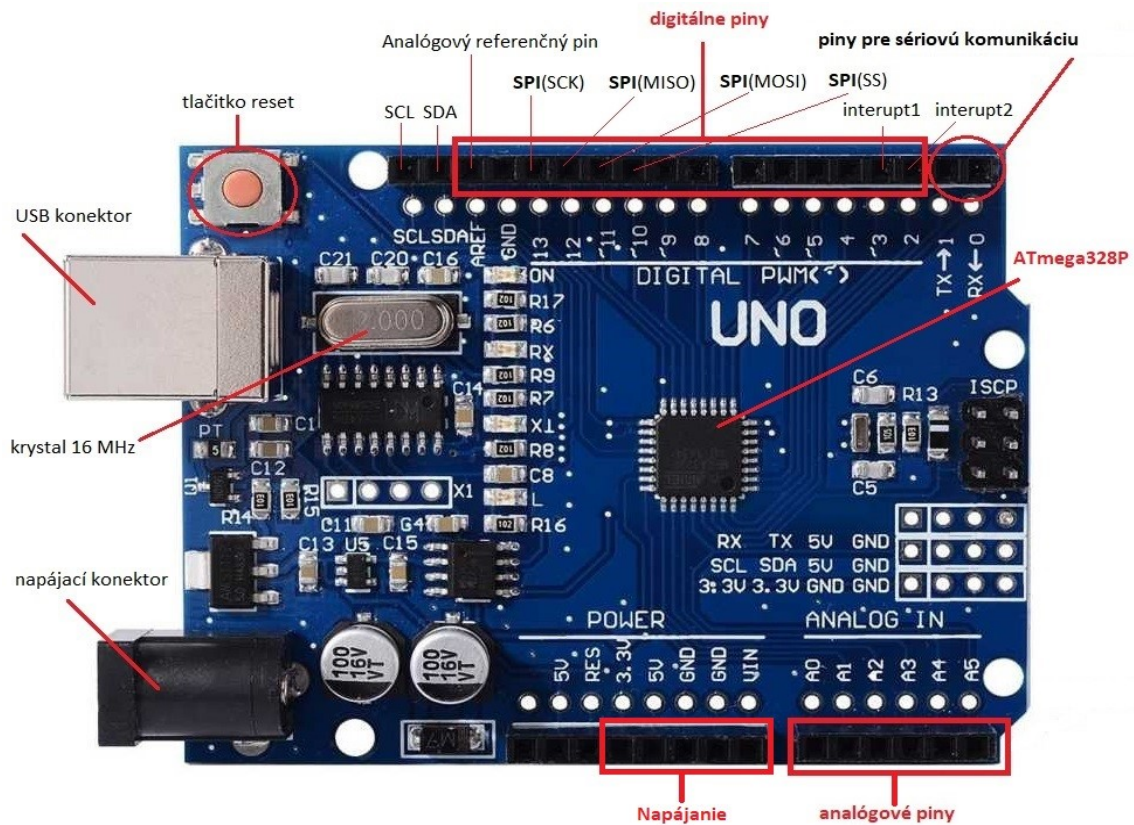
Arduino Fio je určené pre bezdrôtové aplikácie. Táto doska je prispôbená k pripojeniu rôznych bezdrôtových modulov, ktoré sú najčastejšie označene ako *XBEE*. Kvôli kompatibilitě so všetkými *XBee* modulmi tu bolo znížené napätie oproti väčšine ostatných dosiek z 5V na 3,3V. Procesor ATmega328P tu beží na frekvencii 8MHz. [21]



Obrázok 3.5: *Arduino Fio (prevzatý z [23])*

Arduino UNO

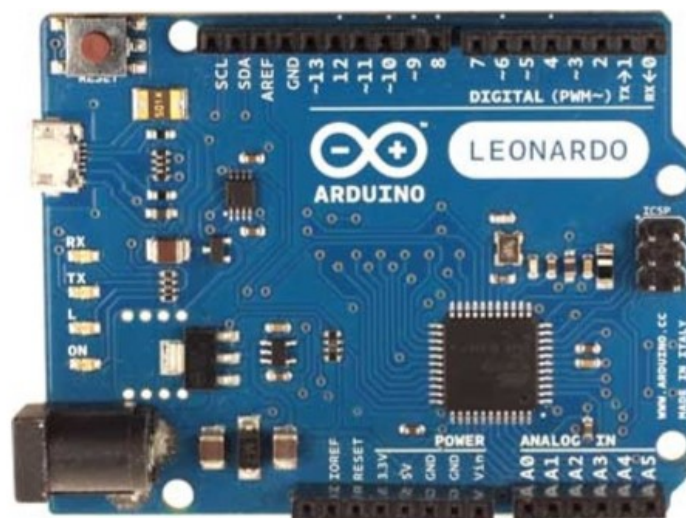
Arduino UNO je v súčasnej dobe asi najčastejšie používaný typ dosky. Je priamym pokračovateľom hlavnej vývojovej línie, ktorá započala prvým Arduinoom so sériovým portom namiesto USB, pokračuje cez Arduino Extreme, NG, Diecimila a Duemilanove až k dnešnému UNO. Doska je obsiahnutá bežným procesorom ATmega328 a má klasické USB a prevodník. Súčasná verzia dosky je *Rev3* alebo *R3*. Táto doska má za následok vývoj niekoľko odlišných variant. Prvá z nich je Arduino Ethernet, ktorá má rovnakú výbavu ako UNO. Namiesto USB portu tu ale nájdeme Ethernet port pre pripojenie k sieti. Príjemná je prítomnosť slotu pre microSD karty. Druhou doskou je Arduino Bluetooth. Ako už názov napovedá, namiesto USB tu nájdeme bluetooth modul pre bezdrôtovú komunikáciu. Veľmi odľahčenou verziou Arduinoa UNO je Arduino Pro. To postráda USB port a je teda nutné ho programovať externým prevodníkom. Je určený skôr k pevnému zabudovaniu do nejakého projektu. [21][23]



Obrázok 2.6: *Arduino UNO*

Arduino Leonardo

Arduino Leonardo designovo naväzuje na Arduino Uno. Líši sa použitým čipom. Tým je ATmega32u4, ktorý bol popísaný už u Arduino Micro. Doska je vhodná do projektov, kde sa môže využiť jeho potenciál fungovať ako zariadenie HID. [23]



Obrázok 2.7: *Arduino Leonardo (prevzatý z [20])*

Arduino Yún

Arduino Yún je model, o ktorom by sa dalo povedať, že sa jedná o modifikáciu Arduino UNO, ale nie je tomu úplne tak. Yún je v týchto doskách skutočný priekopník tým, že je použitý čip ATmega32u4, ktorý má blízko k Arduino Leonardovi. Na tomto čipe beží jadro Arduina. Doska obsahuje rovno dva čipy. Tým druhým je Atheros AR9331. Vo výbave tohoto Arduina je i softwarový most, ktorý zaisťuje komunikáciu medzi čipmi. Doska má zabudovanú podporu Ethernet a Wifi, port USB-A, slot pre kartu micro-SD. Namerané hodnoty sa môžu posielat' priamo na webový server. Spája v sebe silu Linuxu s jednoduchým používaním Arduina. [21]



Obrázok 2.8: *Arduino Yún (prevzatý z [20])*

Arduino Zero

Arduino Zero je jednoduchý a výkonný 32-bitový nasledovník Arduina UNO. Zero je poháňané mikroprocesorom *Atmel S ADM21 MCU* a obsahuje jadro 32-bitové *ARM Cortex M0+*. Jednou z jeho najočakávanejších vlastností je využitie EDBG, ktorý poskytuje plné debugovacie prostredie bez potreby ďalšieho hardwaru. Na doske Zera sa nachádza nový grafický element. Spomínaným elementom je Logo Genuina, čo je sesterský projekt Arduina. Pridaním loga Genuina sa zdôraznila jeho jedinečnosť a zároveň k nemu zjednodušil prístup komunita Arduina. [22]



Obrázok 2.9: *Arduino Zero (prevzatý z [22])*

Arduino Mega256

Arduino Mega je ideálne pre rozsiahlejšie Arduino projekty. Zväčšenie rozmerov prináša priestor pre väčšie a výkonnejšie čipy. Arduino Mega256 je odporúčanou doskou pre 3D tlačiarne a projekty robotiky. [20]



Obrázok 2.10: *Arduino Mega256* (prevzatý z [22])

Arduino Due

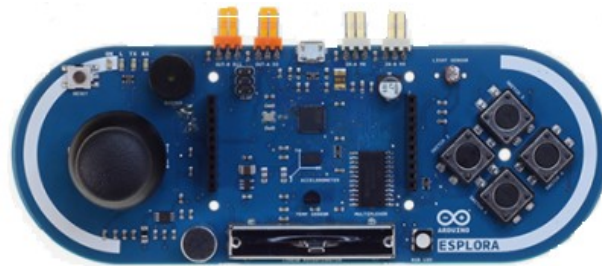
Arduino Due je priamym pokračovateľom Arduina Mega, s tým rozdielom, že beží na ďaleko výkonnejšom procesore, tým je procesor *Atmel SAM3X8E ARM Cortex-M3*. Je to prvá Arduino doska s taktovacou frekvenciou 84MHz, založená na 32-bitovom mikrokontroléri jadra ARM, čo je oproti ostatným doskám s 8-bitmi a maximálne 16MHz veľký skok. Na rozdiel od väčšiny Arduino dosiek Arduino Due doska beží na 3,3V. [23]



Obrázok 2.11: *Arduino Due* (prevzatý z [22])

Arduino Esplora

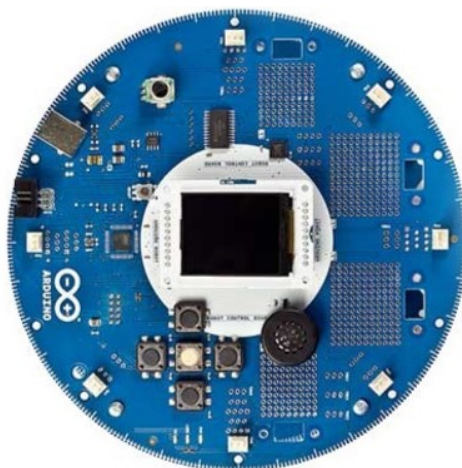
Arduino Esplora je prvá z dosiek, ktorá by sa dala zaradiť do kategórie „hybridných“. Arduino Leonardo je doska so zabudovanými senzormi a ovládačmi. Esplora sa líši od ostatných Arduino dosiek predovšetkým pripravovanými tlačítkami a joystickom priamo na doske. Arduino ďalej ponúka posuvný potenciometer, senzor teploty, akcelometer, mikrofón, snímač osvetlenia, bzučiak alebo piny pre pripojenie LCD displeja. Jedná sa o typ Arduina, s ktorým sa dá vytvoriť samostatný herný set, alebo vlastné konzoly pre hranie hier. [20]



Obrázok 2.12: *Arduino Esplora* (prevzatý z [20])

Arduino Robot

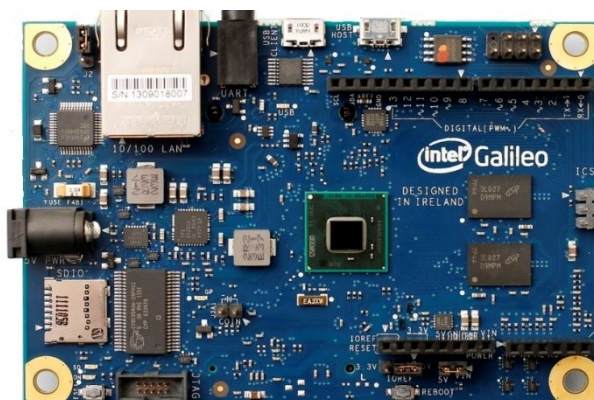
Arduino Robot je prvé oficiálne Arduino na kolesách. Ako už názov napovedá, jedná sa o set pre vytvorenie vlastného chytrého robota. Zaujímavosťou je prítomnosť kompasu. [20]



Obrázok 2.13: *Arduino Robot* (prevzatý z [20])

Arduino Intel Galileo

Arduino Intel Galileo je verzia, ktorá vznikla so spoločnosťou Intel. Doska je založená na procesore Intel®Quart Soc X1000, čo je 32-bitový procesor s frekvenciou 400 MHz. Využitie tejto dosky bude niekde, kde bude potrebný vysoký výpočtový výkon. [21]



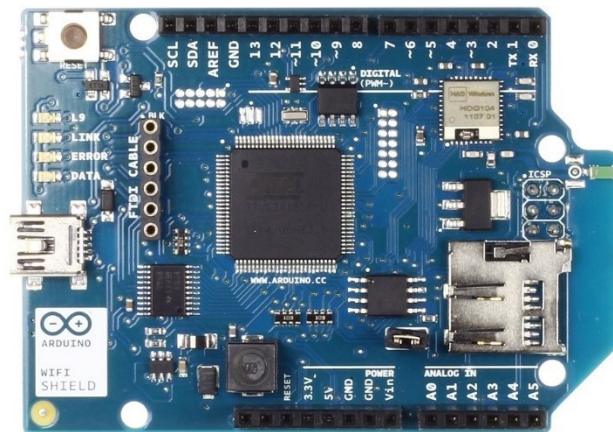
Obrázok 3.14: *Arduino Intel Galileo* (prevzatý z [20])

2.2 Shieldy

Neoddeliteľnou súčasťou Arduina sú práve shieldy, ktoré slúžia k rozšíreniu základnej dosky. Moduly označené logom Arduino sa označujú ako shieldy. Jedná sa o nadstavbu pre Arduino dosky, najčastejšie pre UNO alebo Mega. Rozširujúcich modulov je celá rada, tu sú spomenuté jedny z tých základných. [21]

Arduino WiFi Shield

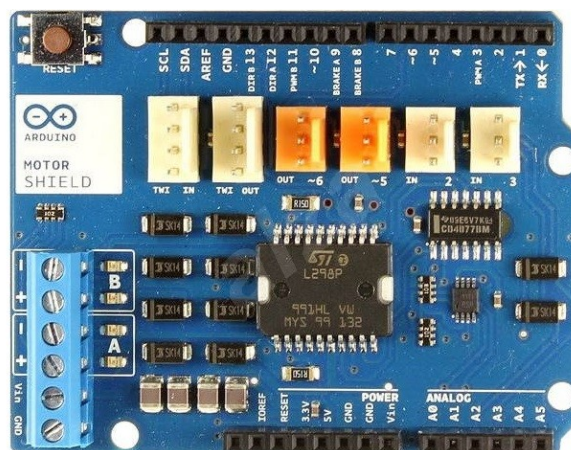
WiFi Shield je rozširujúci modul, ktorý umožní Arduino bezdrátovú komunikáciu cez Wi-Fi. [21]



Obrázok 2.15: *WiFi Shield* (prevzatý z [24])

Arduino Motor Shield

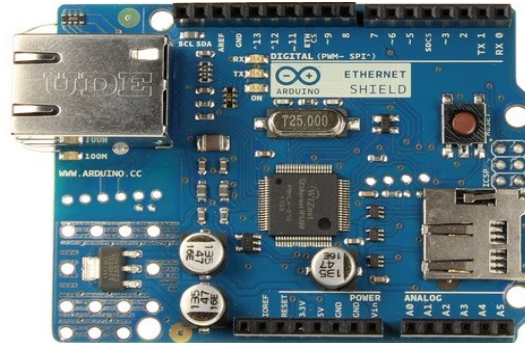
Motor Shield je modul umožňujúci pripojenie motorov. Tie slúžia napríklad pre pohyb robota alebo pre pohon auta na diaľkové ovládanie. [21]



Obrázok 2.16: *Motor Shield* (prevzatý z [25])

Arduino Ethernet Shield

Ethernet Shield je rozširujúca karta, ktorá Arduino pridá možnosť komunikácie prostredníctvom LAN siete. Shield je schopný logovať komunikáciu v sieti alebo namerané hodnoty ukladať na vzdialené úložisko, napríklad na webový server. [21]



Obrázok 2.17: *Ethernet Shield* (prevzatý z [26])

Arduino Relé Shield

Relé Shield je špeciálny typ, ktorý umožňuje spínať ďalší, na Arduine nezávislý obvod. Cez Arduino je možné vytvoriť napríklad spínač na lampu. [21]



Obrázok 2.18: *Relé Shield* (prevzatý z [21])

Arduino GMS Shield

GMS Shield je shield, ktorý dokáže komunikovať prostredníctvom GSM siete. Zvláda GPRS o maximálnej rýchlosti 85,6 *kbps*. Tento modul dokáže odosielať SMSky s nameranými hodnotami. [21]

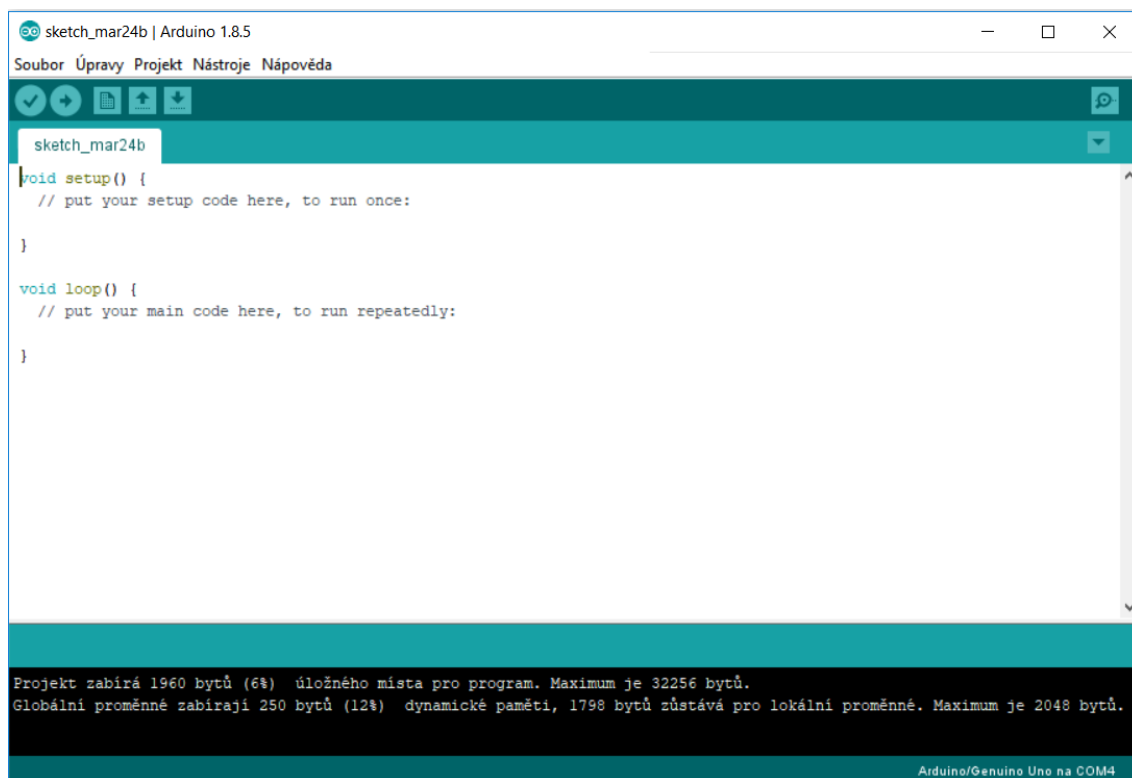


Obrázok 2.19: *GSM Shield* (prevzatý z [21])

2.3 Arduino IDE

Najrozšírenejšie integrované programovacie prostredie pre vytváranie programov pre platformy Arduino. Programovací jazyk je Wiring. Jedná sa v podstate o C/C++ rozšírený o knižnice, uľahčujúce programovanie Arduino. Vývojové prostredie Arduino IDE je napísané v jazyku Java a vychádza z Processingu a z iných open-source. Arduino IDE je dostupné pre všetky bežne využívané operačné systémy Windows, Linux a Mac OS X. Pre všetky tieto platformy je vývojové prostredie možné stiahnuť z oficiálnych stránok.

Arduino IDE si môžeme stiahnuť z oficiálnych stránok <https://www.arduino.cc/>. Pre Windows je najjednoduchšie stiahnuť si ZIP archív, ktorý je po rozbalení plne funkčný. Vývojové prostredie môžeme vidieť na obrázku 2.20.



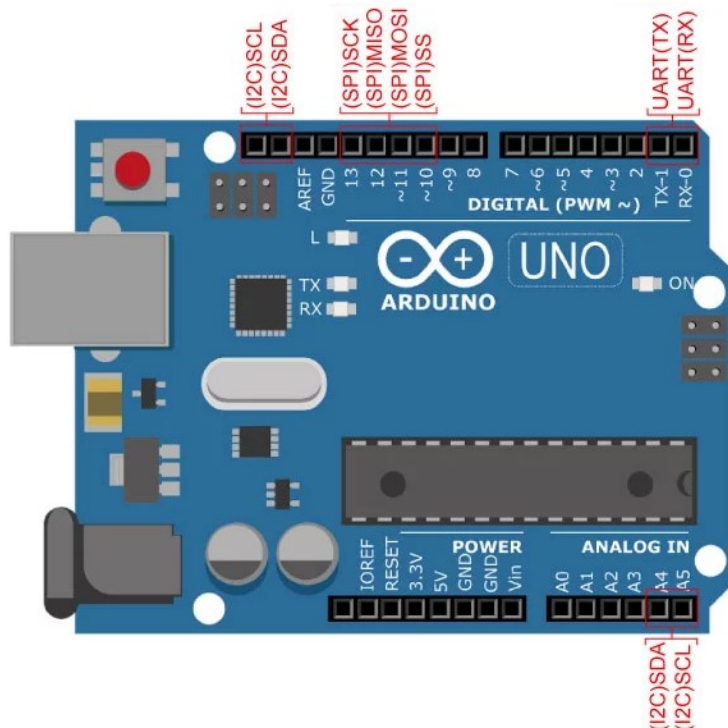
Obrázok 2.20: *Arduino IDE*

Celé prostredie sa skladá z jednoduchého textového editoru, ponuky základného nastavenia a sériového terminálu. Užívateľ v tomto prostredí vytvára tzv. Sketche, súbory s koncovkou *súbor.ino* u starších verzií *súbor.ide*.

Každý program obsahuje 2 základné bloky, *setup()* a *loop()*. Blok programu *setup()* je vykonávaný vždy iba raz po pripojení napájania, stlačení tlačidla reset alebo nahraniu kódu do Arduino. V tejto časti sa vykoná inicializácia a deklarácia premenných. Druhým blokom je *void loop()*, do jeho zložených zátvoriek sa zapisuje kód, ktorý sa bude opakovať neustále dookola až do odpojenia napájania. Ide o nekonečnú smyčku, v ktorej sa vykonáva samotný program. Tieto dve časti musia byť v programe vždy. [23]

2.4 Komunikačné rozhranie

Pre komunikáciu externých obvodov s mikroprocesorom, a prípadne mikroprocesoru s počítačom je potrebné zvoliť vhodné spôsoby komunikácie, pomocou ktorých spolu budú jednotlivé zariadenia komunikovať. Každé Arduino disponuje aspoň jednou sériovou komunikáciou pomocou päť-voltového UART, komunikáciu pomocou SPI a I²C.



Obrázok 2.21: Komunikačné zbernice (prevzatý z [27])

Univerzálny asynchrónny prijímač/vysielač (UART)

UART (Universal Asynchronous Receiver-Transmitter) je jedným z najjednoduchších spôsobov komunikácie medzi dvoma zariadeniami. Na prenos sa používajú dva dátové vodiče - jeden na prijímanie (Rx) a druhý na odosielanie dát (Tx), a jeden spoločný referenčný vodič (zem). Jedná sa o asynchrónny spôsob komunikácie, a preto obsahuje prijímač i vysielač vlastný generátor hodinového signálu, podľa ktorého sa riadi. Pretože je UART univerzálny je možné rýchlosť týchto hodín definovať, rovnako ako rýchlosť bajtov, počet stop bitov, paritné bity, atď.

Ako už bolo zmienené prebieha na vodičoch príjmu (Rx) a vysielaní (Tx), na tomto princípe pracuje i USB kábel, ale je značne rýchlejší. Na doske Arduino UNA sa nachádza pod pinmy 0 a 1. K prepojeniu sa využíva pripojenie do kríža, teda Tx jedného zariadenia sa pripojí na Rx druhého a naopak.

Je to najviac používané, spoľahlivé, funkčné, jednoduché na ovládanie, ale zo všetkých najpomalšie rozhranie. [28]

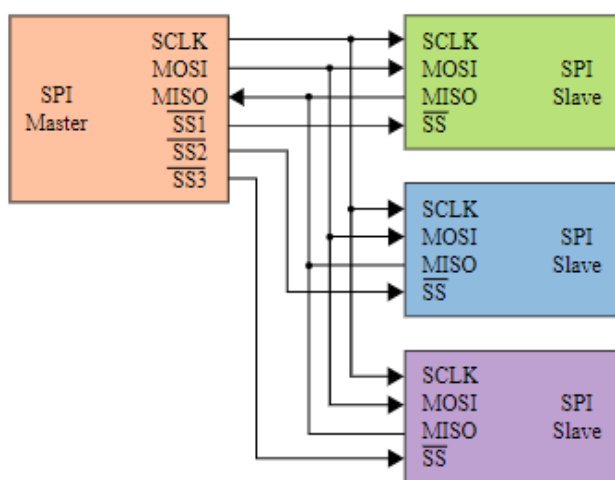
Zbernica SPI

Zbernica Serial Peripheral Interface, skrátene SPI sa hodí k synchronnej komunikácii na krátke vzdialenosti. Podporuje komunikáciu FULL duplex (simultánne odosielanie a prijímanie dát) až do rýchlosti 10 MHz. Používa sa pre prepojenie mikroprocesoru a ďalších obvodov, ktorými môžu byť napr. pamäte EEPROM, displeje, SD karta, A/D prevodníky. Prenos informácií prebieha pomocou štyroch vodičov:

- SCK - hodinový signál, ktorého zdrojom je master a je pripojený ku všetkým zariadeniam. Bežia na ňom hodiny a určuje rýchlosť komunikácie.
- MOSI (Master Out, Slave In) - dátová linka, po ktorej master odosiela dáta.
- MISO (Master In, Slave Out) - dátová linka, z ktorej master prijíma dáta
- SS(n) (slave select) - slúži na výber zariadenia, s ktorými aktuálne prebieha komunikácia. V prípade pripojenia viacerých periférií je potrebné pre každú použiť samostatný vodič tohto typu.

Tabuľka A.3: Piny Arduina pre SPI komunikáciu

Pin	Arduino Uno	Arduino Mega
MISO	12	50
MOSI	11	51
SCK	13	52
SS	10	53



Obrázok 2.22: SPI komunikácia (prevzatý z [28])

Komunikačná sieť vždy obsahuje iba jedno centrálné zariadenie (master), obvykle sa jedná o mikroprocesor, ktoré riadi jednu a viac periférií. Zariadení typu Slave môže byť viac a ich správna spolupráca je zaistená pomocou adresovacieho pinu SS. Keď Master prepne SS pin do logickej 0 môže komunikovať s daným Slavom. Ostatné zariadenia sa umlčia. Pre ukončenie komunikácie s daným Slavom Master nastaví príslušný pin SS do *log* 1 a potom môže začať komunikovať s ďalším zariadením podľa rovnakej logiky. [28]

Zbernica I²C

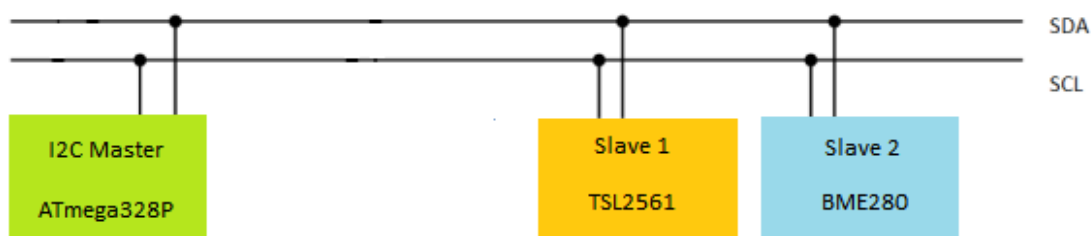
Pod názvom Inter-Integrated Circuit sa ukrýva multi-masterová zbernica, ktorú vyvinula firma Philips. Ide o internú sériovú dátovú zbernicu, ktorá sa využíva pri komunikácii v rámci jedného zariadenia. Zbernicu I²C je možno nájsť vo výbave mikroprocesorov, pamätí, obvodu reálneho času, A/D prevodníku, D/A prevodníku, senzorov.

Táto zbernica beží na princípe *Master–Slave*. Komunikácia nie je duplexná, preto pre komunikáciu stačia iba 2 vodiče:

- SDA (serial data) - slúži k vedeniu dát,
- SCL (serial clock)- slúži k vedeniu hodinového signálu.

Tabuľka A.4: *Piny Arduina pre I²C komunikáciu*

<i>Pin</i>	<i>Arduino UNO, Arduino Pro Mini</i>	<i>Arduino Mega, Arduino Due</i>	<i>Arduino Leonardo, Arduino Pro Micro</i>
SDA	A4	20	2
SCL	A5	21	3



Obrázok 2.23: *I²C komunikácia*

Komunikácia začne, keď Master vyšle START bit a následne adresu volaného zariadenia. U všetkých zariadení je prijatá adresa porovnávaná s nastavenou adresou daného zariadenia, a pokiaľ sa zhoduje, tak je vyslaný potvrdzujúci bit. Každé vysielanie začína START bitom a následne bitom potvrdzujúcim, či bude Master vysielat' alebo čítať. K ukončeniu komunikácie je potrebné vysielanie STOP bitu.

Komunikácia je synchronizovaná pomocou hodinového signálu a môže dosahovať rýchlosť až 400 kHz. Adresovanie prebieha pomocou pridelených 7 bitovej či 10 bitovej adresy. Zoznam adries, ktoré používajú senzory:

- 0x39 - TSL 2561,
- 0x76 - BME 280.

I²C zbernica je vhodná pre dosky s mnohými zariadeniami pripojenými na zbernicu. To pomáha znížiť náklady a zložitosť obvodu. Nevýhodou tohto typu komunikácie je, že Master môže komunikovať iba s jedným zariadením, iba jedným smerom a Slave si Mastra nezavolá. [28]

3 Vizualizačný nástroj Grafana

Ako vizualizačný systém bola zvolená Grafana, analytická platforma pre monitorovanie a vizualizáciu dát pomocou grafov. Je to jeden z najrozšírenejších a najprepracovanejších vizualizačných nástrojov, určených pre zobrazovanie časovo usporiadaných a číselných dát.

Grafana je open-source softvér pre analýzu časových radov. Grafana umožňuje zobrazovať data v časovom slede, čo je ideálne z hľadiska dlhodobého sledovania a diagnostiky. Je používaná spoločnosťami napr. Intel či Vimeo. Je dostupná ako webová aplikácia pre všetky bežné operačné systémy a aj ako cloudová služba.

Jednou z výhod grafov vygenerovaných pomocou Grafany je ich užívateľská prívetivosť, a to ako z pohľadu grafiky, tak z pohľadu možnosti interaktívne s nimi pracovať. Grafana poskytuje možnosť vývoja externých pluginov, tak vznikla škála rôzne orientovaných rozšírení, ktoré je možné doinštalovať a získať tak veľa ďalších možností. [29] [33]

3.1 Inštalácia a konfigurácia

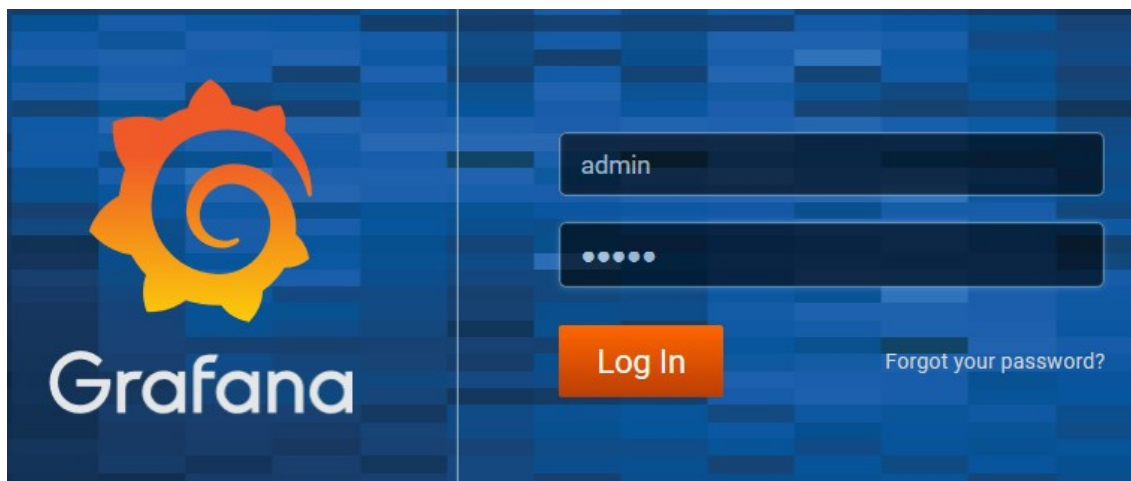
Inštalácia monitorovacieho systému Grafana sa líši na rôznych operačných systémoch. V prostredí Linuxu existujú pripravené balíčky, ktoré odporúčam stiahnuť priamo z oficiálnych stránok <https://grafana.com/grafana>. [30]

Vo výpise B.4 sú potrebné príkazy pre inštaláciu a spustenie. Všetky závislosti by mali byť priamo v balíčkoch a inštaláciu je možné vykonať programom *dpkg*, správcom balíčok pre Linux. Pre program nie je potrebná žiadna konfigurácia pred spustením. Následne sa aktualizuje úložisko a nainštalujete Grafana pomocou nasledujúcich príkazov. Ďalším príkazom sa spustí proces *grafana – server* ako užívateľ *grafana*, ktorý bol vytvorený počas inštalácie.

Výpis B.4: *Inštalácia Grafany*

```
$ wget https://dl.grafana.com/oss/release/grafana_5.4.2_amd64.deb
$ sudo dpkg -i grafana_5.4.2_amd64.deb
$ sudo apt-get update
$ sudo apt-get install grafana
$ sudo service grafana-server start
```

Po inštalácii je Grafana ihneď pripravená k použitiu, teda k nastaveniu zdroju dát a plochy dashboardu. V prehliadači prejdeme na adresu: *http://localhost: 3000*. Defaultný port je 3000 a prihlasovacie údaje sú *admin / admin* (Obr. 3.1).



Obrázok 3.1: Grafana - prihlásenie

3.2 Data Sources

Aby mohla Grafana vytvárať vizualizáciu, je potrebné najprv nadefinovať dátové zdroje. Data Sources sú zdroje z ktorých sa načítajú a zobrazujú data. Grafana podporuje prevažnú väčšinu zmieňovaných úložísk pre časovo usporiadané dáta, ako sú napr. Graphite, Prometheus, Elasticsearch, InfluxDB, OpenTSDB, AWS CloudWatch. Ako zdroj dát bola nastavená InfluxDB databáza. Na obr. 3.2 je zobrazené nastavenie InfluxDB databázy ako zdroj dát.

V Grafane je možné kombinovať viacero zdrojov dát a to ako naprieč grafmi, tak i v jednotlivých grafoch. Pomocou rozličných užívateľských modulov je možné Grafanu rozšíriť o mnohé funkcionality či ďalší zdroj dát. V Grafane je možné pustiť takzvaný real-time mód, ktorý začne zobrazené grafy aktualizovať o nové dáta v zadanom intervale. Ďalšou užitočnou vlastnosťou je využitie metadát pre filtrovanie hodnôt v zobrazených grafoch. [31]

Okrem napojenia na tieto zdroje poskytuje aj množstvo pluginov umožňujúcim pripojenie na ďalšie dátové zdroje. Vizualizácia potom predstavuje definíciu dotazu a zobrazenie získaných dát. Konfigurácia dátového zdroja je základ pre vizualizáciu dát a prebieha za behu Grafany, ktorá si dátový zdroj uloží do svojej vnútornej databázy. Aby sme zabránili jeho strate je možné nakonfigurovať databázu externú. Druhou možnosťou je výstavba obrazu, ktorý pri každom naštartovaní kontajneru, po spustení Grafany, sám importuje preddefinované dátové zdroje. Tie by mohli byť špecifikované pomocou premenných prostredia alebo importované z pripojenej konfiguračnej mapy obsahujúce pred-pripravené súbory vo formáte JSON s ich definíciami.

Podobne ako k dátovým zdrojom je možné tiež pristupovať k vizualizačným sadám. Zatiaľ čo dátové zdroje je ešte možné nakonfigurovať ručne v prípadoch, kedy využívame jeden alebo dva, a kontajner je resetovaný len raz za dlhú dobu, nejde to isté povedať o tabuľkách. Tie predstavujú definície dotazov a nastavení vizualizácií, a ich ručné konfigurácie zaberú mnoho času. I tu sa opäť ponúka niekoľko možností ako pristupovať k tomuto problému. Rovnako tak ako v prípade datových zdrojov je možné zvoliť zálohovanie v externej databázi, to nám však neprinesie jednoduchú prenositeľnosť. [32]

Settings

Name	GRAFANA	i	Default	<input type="checkbox"/>
------	---------	-------------------	---------	--------------------------

HTTP

URL	http://localhost:8086	i
Access	Server (Default)	Help ▶
Whitelisted Cookies	Add Name	i

Auth

Basic Auth	<input checked="" type="checkbox"/>	With Credentials	i	<input type="checkbox"/>
TLS Client Auth	<input type="checkbox"/>	With CA Cert	i	<input type="checkbox"/>
Skip TLS Verify	<input type="checkbox"/>			

Basic Auth Details

User	ea
Password

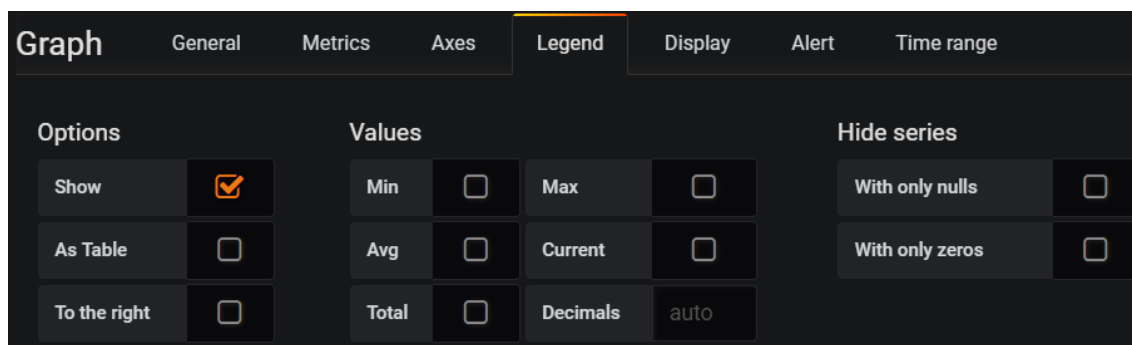
InfluxDB Details

Database	GRAFANA		
User	ea	Password

Obrázok 3.2: Grafana - zdroj dát

3.3 Graph Panel

Graph panel ponúka bohaté možnosti nastavenia grafu. Nastavenia sú dostupné po kliknutí na názov grafu. Obr. 3.3 zobrazuje možnosti Graph panela, ktoré sú nižšie popísané.



Obrázok 3.3: Grafana - Graph Panel

- Karta základných nastavení (General) - obsahuje možnosti úpravy názvu a rozmerov grafu.
- Metriky (Metrics) - definuje aké postupnosti dát a zdroje sa zobrazia v grafe.
- Osy (Axes) - táto časť umožňuje nastavenie osí, mriežok a legend grafu. Polia *Left Y* a *Right Y* slúžia pre nastavenie osi Y. Nastaviteľná je jednotka osi, štítok a minimálna a maximálna hodnota.
- Legenda (Legend) - môže byť zobrazená ako skrytá. Doplnkové informácie k legende grafu môžu zobrazovať celkový počet hodnôt vrátených pre danú metriku, poslednú hodnotu metriky, minimálnu, maximálnu alebo priemernú hodnotu metriky.
- Štýly zobrazenia (Display):
 - Hranica (Threshold) - pridanie ľubovoľných čiar alebo sekcií do grafu pre zvýraznenie prekročenia určitej hodnoty.
 - Možnosti diagramu (Draw Modes) - zobrazenie hodnôt ako stĺpcový diagram (bar), ako čiarový graf (line graph) alebo ako postupnosť bodov (points graph).
 - Možnosti čiar (Mode Options) - nastavenie výplne plochy grafu, šírky čiar, prípadne správanie grafu pri zobrazení NULL hodnôt, či schodové zobrazenie po sebe nasledujúcich bodov.
 - Zobrazenie viacerých postupností - môžu byť zobrazené ako skupina. Možnosť *Stack* umiestni série na seba. Možnosť *Percent* vykreslí každú postupnosť ako v percentách ku celkovej hodnote všetkých postupností. [34]

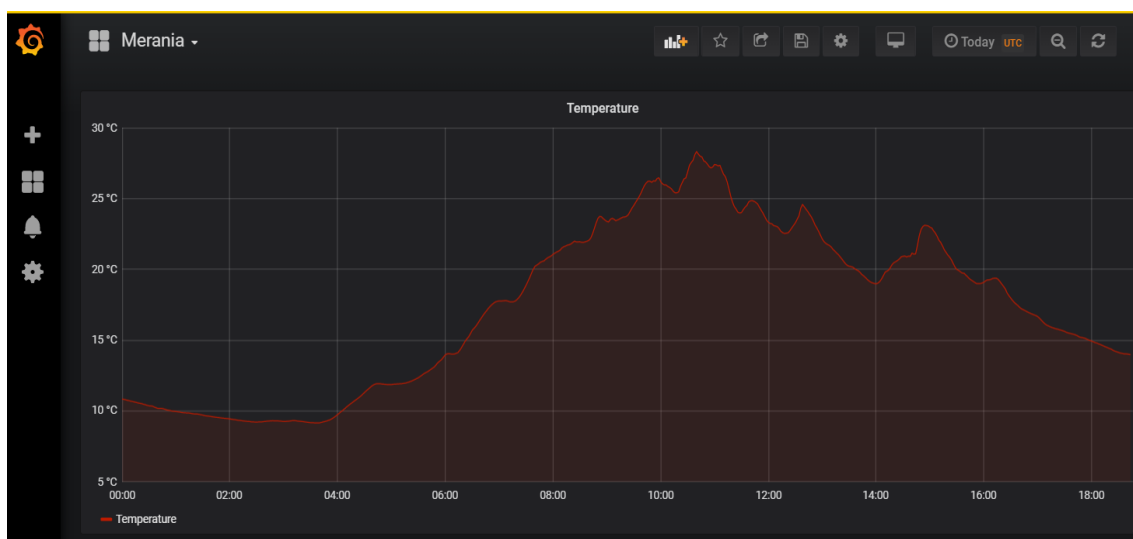
Grafana poskytuje množstvo ďalších nastavení, ako definícia offsetu, zafarbenie alebo automatická vizualizácia meranej jednotky. Ďalšie vizualizácie môžu byť opäť doinštalované ako modul.

Taktiež je možné urobiť import panelov súčasťou spusteného procesu kontajnera výstavbou nového obrazu. [34]

3.4 Dashboard

Grafana je rozdelená do niekoľkých častí, kde hlavný z nich je tzv. dashboard, čo je zobrazovací priestor, kde je možné umiestňovať panely v rôznych typoch, množstvách alebo veľkostiach. Dashboard udržuje pohromade nastavenia grafov a ich množinu predkonfigurovanú užívateľom. Grafana umožňuje vytváranie si vlastných dashboardov, do ktorých je možné vkladať grafy či dlaždice s konkrétnymi hodnotami. Počet týchto obrazoviek pre užívateľa nie je obmedzený a je možné ich aj exportovať alebo importovať niektoré z verejne dostupných. [33]

Užívateľský panel dashboard je členený do riadkov. Do riadkov sú umiestnené jednotlivé vizualizačné prvky. Na obr. 3.4 je vidieť webové rozhranie monitorovacieho systému Grafana, v ktorom je zobrazený graf s nameranými hodnotami teploty vzduchu.



Obrázok 3.4: Grafana - ukážkový dashboard

Šablóny

Vytváranie šablón poskytuje pohodlnejšiu kontrolu nad zobrazovaním obsahu dashboardu. Využíva premenné do ktorých je možné uložiť napr. názov aplikácie, serveru, metriky alebo atribútu premennej. Tie sú dostupné na vrchnej časti dashboard v dropdown liste a pomocou neho je možné jednoducho meniť obsah zobrazovaný na dashboard. [34]

Anotácie

Pomocou anotácií je možné označiť bod v grafe. Anotácia pri prejdení kurzorom zobrazí bližšie informácie. Anotácie je možné vytvoriť cez annotation queries, kde je na výber zdroj dát, meno a komentár. [34]

Medzi nezanedbateľné funkcie tiež patrí možnosť definície systému upozornení. Podľa nadefinovaných pravidiel serverová časť Grafany pravidelne spúšťa dotazy a vyhodnocuje získané výsledky. V prípade porušenia pravidiel upozorní užívateľa (napríklad zaslaním emailu).

4 Návrh riešenia automatizovaného zberu dát

V tejto kapitole budú uvedené podrobnosti o tom, ako prebiehala implementácia tohto projektu od zapojenie senzorov, cez nastavenia prostredia pre vývoj, naprogramovanie skriptu pre zber dát, až po vykresľovanie grafov.

4.1 Realizácia modulu

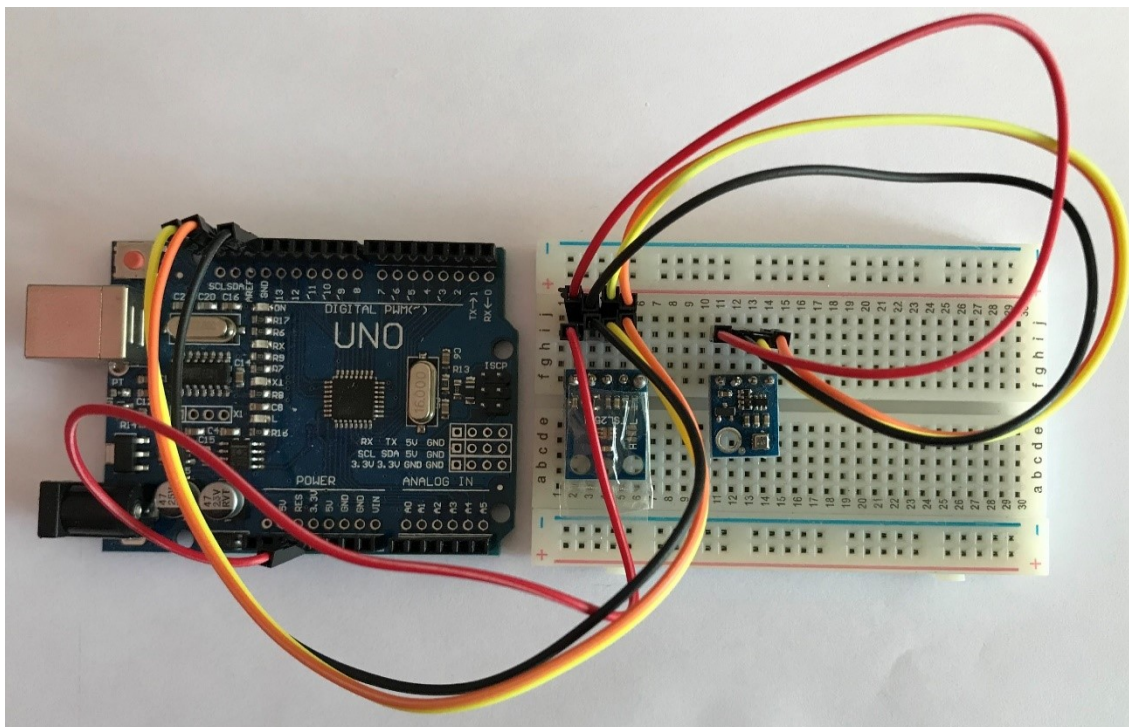
Na obrázku 4.1 je ilustratívne zostavenie senzorov s doskou Arduino. Táto zostava mi umožnila snímať intenzitu osvetlenia, teplotu, vlhkosť a tlak.

Arduino UNO pravdepodobne patrí medzi najpoužívanejšiu dosku pre Arduino. Zvolenie tohto mikrokontroléra je vhodné najmä kvôli jeho kompatibilite naprieč celým spektrom senzorov a rôznych zariadení. Medzi jeho výhody patrí pomerne nízka cena a variabilita zapojení.

Po preskúmaní dostupných senzorov boli zvolené senzory TSL 2561 a BME 280.

Senzor, pomocou ktorého budeme zisťovať teplotu okolia, vlhkosť vzduchu a barometrický tlak je digitálny senzor BME 280. Voľba jedného integrovaného senzora padla pre tento senzor, pretože zahŕňa viaceré samostatné jednotky pre meranie rôznych veličín v jednom obvode. Je každopádne výhodou z hľadiska výkonovej spotreby a malej veľkosti.

Svetelný senzor TSL 2561 budem v mojom návrhu používať pre jeho nízku spotrebu energie. Veľkou výhodou je, že tento svetelný senzor dokáže vylúčiť z meraného spektra ako infračervené, tak i ultrafialové žiarenie a pre svoje fungovanie taktiež nepotrebuje žiadne komplexné algoritmy.



Obrázok 4.1: Zostava základnej dosky Arduino UNO a použitých senzorov

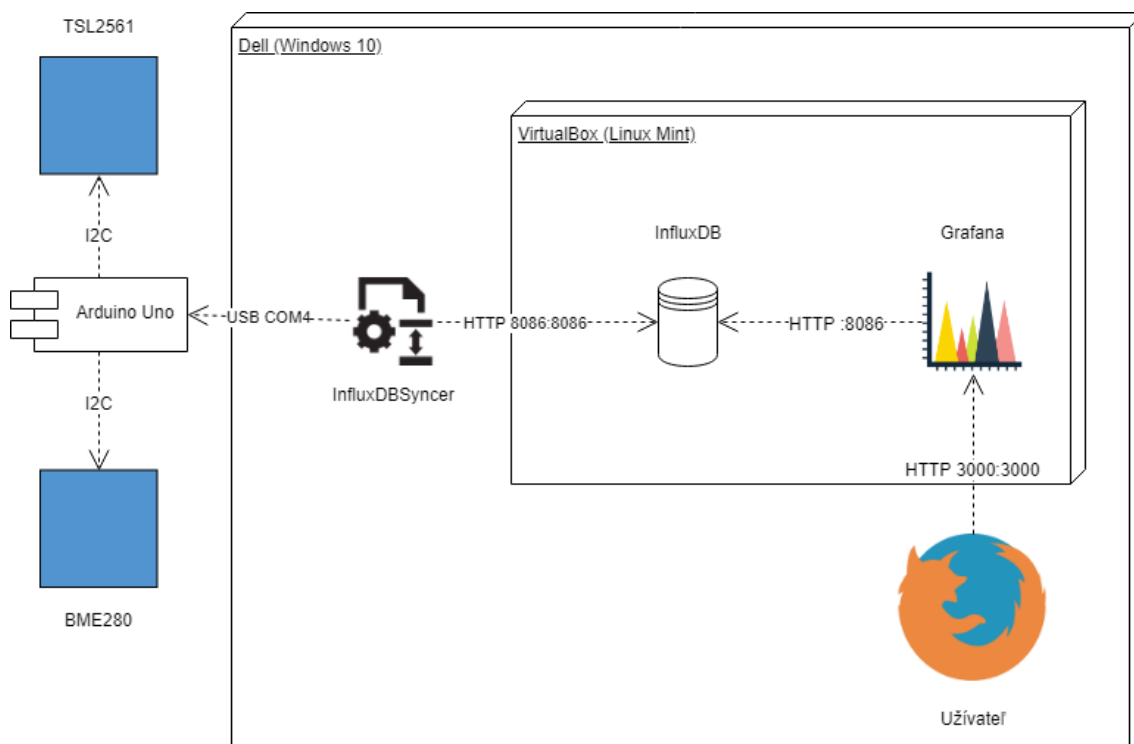
4.2 Návrh riešenia

Obrázok 4.2 zachytáva model nasadenia. Ako operačný systém som používala Windows 10. Ďalej som použila virtualizačný nástroj VirtualBox (Linux Mint), ktorý bol pre moje potreby dostačujúci. Virtuálny stroj som vytvorila pre InfluxDB a Grafanu. InfluxDB i Grafana poskytujú inštalateľné balíčky pre Linux.

Mikrokontrolér Arduino UNO má k sebe pripojené 2 senzory cez I²C zbernicu. Komunikácia medzi mikrokontrolérom a notebookom prebieha prostredníctvom USB, na ktorom je sériový prevodník do sériového portu (na notebooku ako COM4).

Prostredníka medzi existujúcimi riešeniami tvorí python skript InfluxDBSyncer. Tento skript získava dáta z mikrokontroléra Arduino a ukladá ich cez http protokol do InfluxDB databázy, odkiaľ si ich môže nahráť Grafana pre vykreslenie grafov.

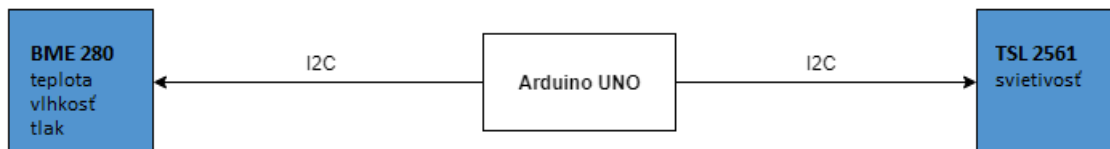
Cez prehliadač je možné pozrieť sa do Grafany, ktorá si načíta dáta z InfluxDB databázy a vykreslí z nich grafy.



Obrázok 4.2: Model monitorovacieho nasadenia

4.3 Zapojenie senzorovej časti

Usporiadanie senzorovej časti zariadenia je znázornené v blokovej schéme na obr. 4.3. Základ celej časti tvorí mikroprocesor *Atmega328P* obsiahnutý vo vývojovej doske Arduino UNO. K mikrokontroléru sú pomocou zbernice I²C pripojené senzory BME 280 a TSL 2561. Modul BME 280 má čidlo teploty, vlhkosti a tlaku a modul TSL 2561 má na sebe čidlo svetla.

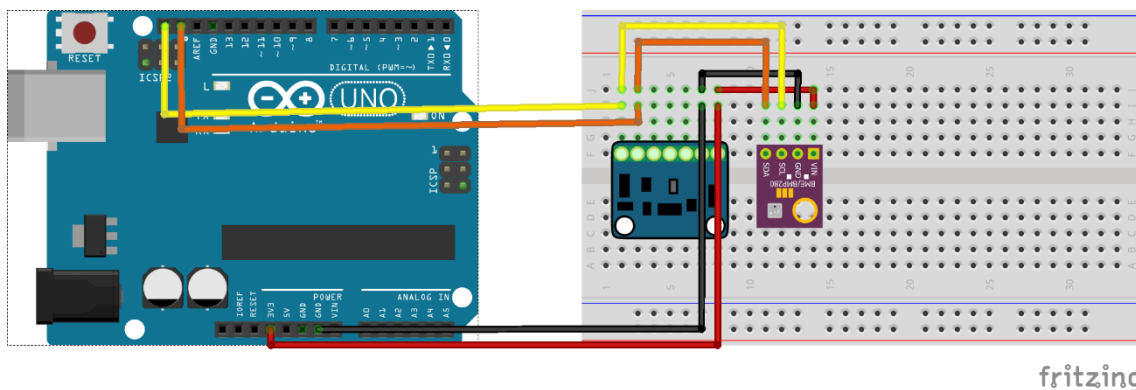


Obrázok 4.3: *Bloková schéma*

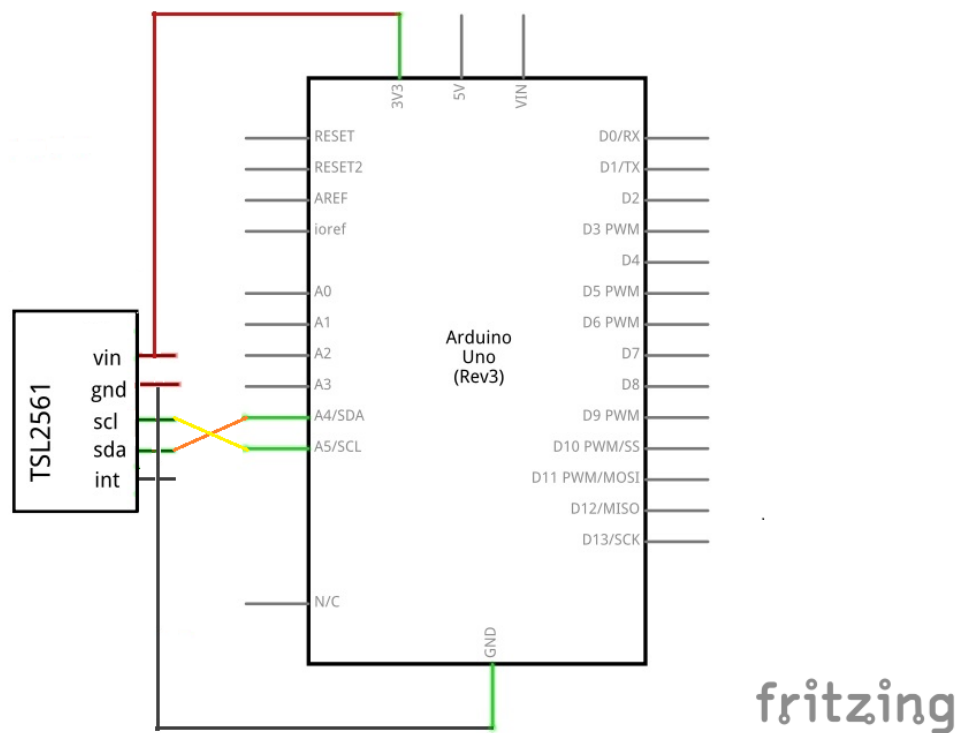
Senzory bolo potrebné zapojiť k modulu Arduino UNO, aby bolo zaistené ich napájanie a prenos nazbieraných dát. Zapojenie pre oba senzory bude rovnaké, pretože používajú zhodnú komunikáciu I²C.

Prvým senzorom, ktorý bolo potrebný zapojiť k modulu Arduino je svetelný senzor TSL2561. Senzor pracuje pri operačnom napätí 3,3 V, ale je vybavený i napäťovým regulátorom z 5V na 3V. Pri meraní spotrebúva podľa technickej špecifikácie 0,5 mA a v kľudovom stave približne 15 uA. Na jednej I²C zbernici je možné mať pripojené až 3 rovnaké senzory, nakoľko je možné pomocou vhodnej logickej hodnoty nastaviť 3 rôzne adresy.

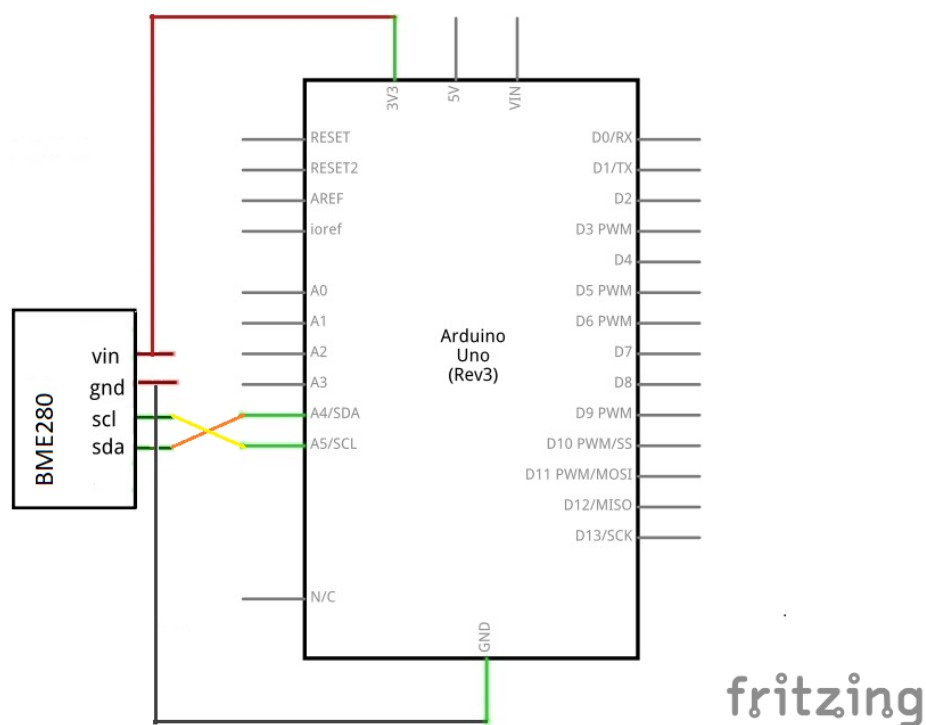
Druhým senzorom, ktorý bolo treba správne zapojiť je BME280. Pre úspešné zapojenie je nutné prepojiť štyri piny. Prepojí sa VIN s 3,3 V Arduina, GND so zemou Arduina, SCL s pinom A5 a SDA s pinom A4. Hodinový pin SCL a dátový pin SDA je nutné zapojiť na I²C piny použitej Arduino dosky.



Obrázok 4.4: *Grafická schéma pripojenia s Arduino*



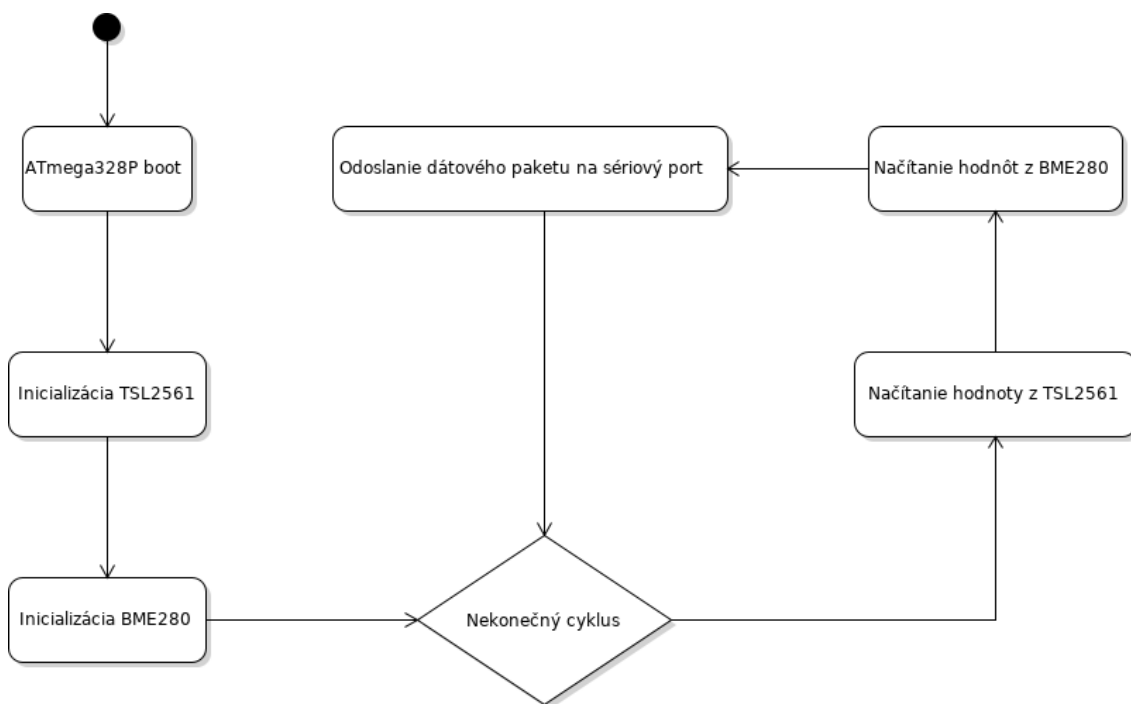
Obrázok 4.5: Elektrická schéma zapojenia senzoru TSL 2561



Obrázok 4.6: Elektrická schéma zapojenia senzoru BME 280

4.4 Programové vybavenie

Program pre mikrokontrolér Arduino UNO je napísaný v jazyku Wiring vo vývojovom prostredí Arduino IDE. Preložený súbor je potom priamo z toho IDE pomocou bootloderu nahraný do mikroprocesoru. Po reštarte zariadenia sa vykoná inicializácia všetkých potrebných registrov a komunikačného modulu. Potom sa vykoná meranie veličín svetla, teploty, tlaku a vlhkosti. Na obrázku 4.7 je zobrazený popis programového vybavenia aplikácie pomocou vývojového diagramu.



Obrázok 4.7: Vývojový diagram funkčnosti aplikácie

Program pre riadenie Arduina sa skladá z niekoľkých častí. Pre správnu funkciu kódu je nutné stiahnuť a nainštalovať knižnice pre senzory BME 280 a TSL 2561 (Výpis B.5).

Výpis B.5: *Potrebné knižnice pre použité senzory*

```
#include <Adafruit_TSL2561_U.h>
#include <Adafruit_BME280.h>
```

Kód obsahuje na svojom začiatku pripojenie nainštalovaných knižníc, definovanie adries pre komunikáciu po I²C rozhraní a potom inicializáciu senzorov z pripojených knižníc. Senzor TSL 2561 má definovanú adresu 0x39 a senzor BME 280 má definovanú adresu 0x76.

Potom nasleduje prvý cyklus *setup()*, ktorý prebehne iba raz po spustení Arduina. Ako prvé sa nastaví komunikácia po sériovej linke a následne sa zaháji komunikácia so senzorom TSL 2561 a následne so senzorom BME 280. Pokiaľ by zlyhalo nastavenie komunikácie, vypíše

sa chybová hláška a program sa zastaví. Vo výpise B.6 je ukážka zahájenia komunikácie so senzorom TSL 2561.

Výpis B.6: *Programové vybavenie - cyklus setup()*

```
void setup()
{
    Serial.begin(9600);
    Serial.println();
    if (!tsl.begin())
    {
        Serial.print("Chyba: senzor TSL2561 nie je pripojeny!");
        while (1);
    }
    tsl.enableAutoRange(true);
    tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_402MS);
}
```

Po vykonaní tohto cyklu prejde program k hlavnému cyklu *loop*, ktorý beží po zvyšok času, pokiaľ nie je program nútený vykonať niečo iné. Nekonečný cyklus obsahuje vypísanie všetkých dostupných údajov zo senzorov, nasledované pauzou po dobu 0,5 sekúnd.

Výpis B.7: *Programové vybavenie - cyklus loop()*

```
void loop()
{
    sensors_event_t event;
    tsl.getEvent(&event);
    if (event.light)
    {
        Serial.print("luminosity:");
        Serial.println(event.light);
    }
    delay(500);
}
```

V nasledujúcom pseudokóde (Výpis B.8) je zhrnutá funkcionálna programového vybavenia.

Výpis B.8: *Programové vybavenie - pseudokód*

```
1  inicializuj senzor svetla
2  inicializuj senzor teploty/tlaku/vlhkosti
3  inicializuj seriový port
4
5  if senzor svetla nie je inicializovaný
6    vypis chybu do serialu a zastav program
7
8  nakonfiguruj svetelný senzor
9
10 if senzor senzor teploty/tlaku/vlhkosti nie je inicializovaný
11   vypis chybu do serialu a zastav program
12
13 while true
14   do
15     zmeraj svetelnosť
16
17     if meranie sa podarilo
18       vypis hodnotu do serialu
19     end
20
21     zmeraj teplotu
22
23     if meranie sa podarilo
24       vypis hodnotu do serialu
25     end
26
27     zmeraj vlhkosť
28
29     if meranie sa podarilo
30       vypis hodnotu do serialu
31     end
32
33     zmeraj tlak
34
35     if meranie sa podarilo
36       vypis hodnotu do serialu
37     end
38
39     pozastav program na 500 milisekúnd
40   end
```


4.5 Zber a uloženie dát

Zber dát

Realizácia zberu dát je sprostredkovaná pomocou jednorazového skriptu InfluxDBSyncer. InfluxDBSyncer je skript, napísaný v jazyku Python. V pseudokóde vo výpise B.9 je zhrnutá implementácia skriptu. Po pripojení obvodu na zbernicu USB operačný systém rozozná zariadenie ako virtuálny sériový port typu COM4. InfluxDBSyncer sa periodicky dotazuje Arduina UNA na aktuálne hodnoty z monitoring systému a zapíše ich do InfluxDB databázy. Na obrázku 4.8 je ukážka debug hlášky z terminálu pre jednotlivé namerané hodnoty.

Výpis B.9: *Implementácia InfluxDBSyncer - pseudokód*

```

1  inicializuj pripojenie do InfluxDB na adrese localhost:8086
2  otvor seriový port COM4
3
4  while true
5  do
6      nactaj 1 riadok zo serialu
7
8      if riadok zacína slovom chyba
9      do
10         vypis chybu na obrazovku
11         ukonci program
12     end
13
14     rozdel riadok na názov merania a hodnotu
15     konvertuj hodnotu ako desatinne číslo
16     vytvor datový paket s názvom merania a hodnotou pre InfluxDB
17     odosli datový paket do InfluxDB
18
19     if odoslanie sa podarilo
20     do
21         vypis na obrazovku debug hlášku s názvom merania a hodnotou
22     end
23     else
24     do
25         vypis na obrazovku chybovú hlášku
26     end
27 end

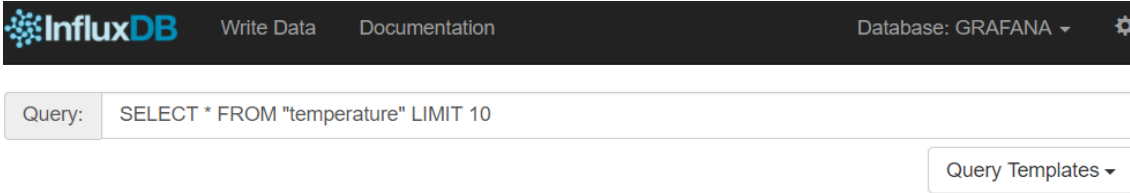
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
			2019-04-22 05:15:33.088703 luminosity: 910.0 Success
			2019-04-22 05:15:33.120906 temperature: 9.0 Success
			2019-04-22 05:15:33.134570 humidity: 57.49 Success
			2019-04-22 05:15:33.152139 pressure: 98574.3 Success

Obrázok 4.8: *Výpis debug hlášky z terminálu*

Uloženie dát

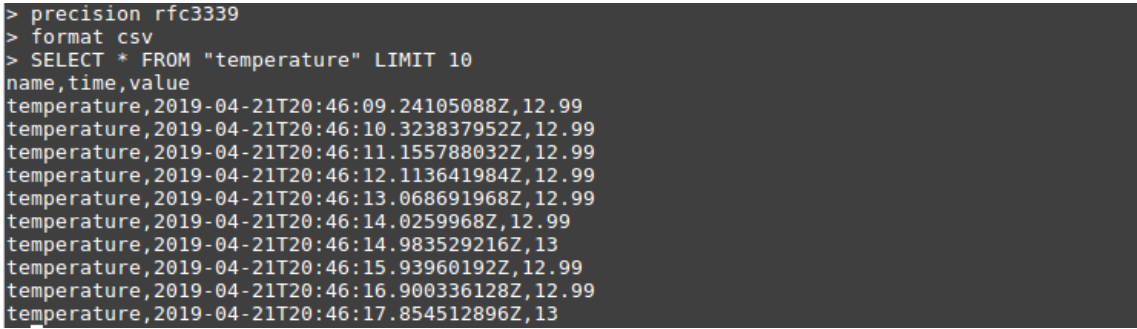
Ukladanie dát je realizované pomocou time series databázy InfluxDB. Dátový model InfluxDB sa skladá z meraní. Merania sú podobné tabuľkám, ktorých môže byť až tisíce. Vo vnútri meraní je množina oddelených časových rád, ktoré sú definované množinou tagov. Tagy sú dvojice (kľúč-hodnota) reprezentujúce metadáta. Všetky tagy sú automaticky indexované. Na obrázku 4.9 je ukážka webového rozhrania, kde vidíme zobrazenie prvých desiatich hodnôt teploty po sekundách. Na obrázku 4.10 je ukážka tých istých hodnôt pomocou rozhrania príkazového riadku.



The screenshot shows the InfluxDB web interface. At the top, there's a navigation bar with 'InfluxDB', 'Write Data', 'Documentation', and 'Database: GRAFANA'. Below this is a query input field containing 'SELECT * FROM "temperature" LIMIT 10'. To the right of the input is a 'Query Templates' dropdown. Below the query input, the title 'temperature' is displayed. Underneath, a table shows the results of the query. The table has two columns: 'time' and 'value'. It contains 10 rows of data, each representing a timestamp and a temperature value.

time	value
2019-04-21T20:46:09.24105088Z	12.99
2019-04-21T20:46:10.323837952Z	12.99
2019-04-21T20:46:11.155788032Z	12.99
2019-04-21T20:46:12.113641984Z	12.99
2019-04-21T20:46:13.068691968Z	12.99
2019-04-21T20:46:14.0259968Z	12.99
2019-04-21T20:46:14.983529216Z	13
2019-04-21T20:46:15.93960192Z	12.99
2019-04-21T20:46:16.900336128Z	12.99
2019-04-21T20:46:17.854512896Z	13

Obrázok 4.9: GUI - ukážka prvých desiatich hodnôt v InfluxDB



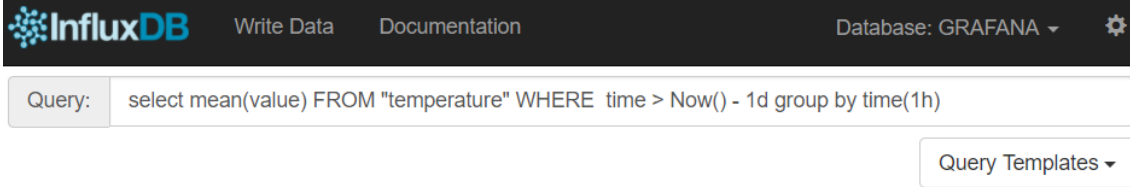
The screenshot shows the InfluxDB CLI interface. It displays the output of the same query as the GUI. The output is in CSV format, with columns 'name', 'time', and 'value'. It shows 10 rows of data, each representing a timestamp and a temperature value.

```
> precision rfc3339
> format csv
> SELECT * FROM "temperature" LIMIT 10
name,time,value
temperature,2019-04-21T20:46:09.24105088Z,12.99
temperature,2019-04-21T20:46:10.323837952Z,12.99
temperature,2019-04-21T20:46:11.155788032Z,12.99
temperature,2019-04-21T20:46:12.113641984Z,12.99
temperature,2019-04-21T20:46:13.068691968Z,12.99
temperature,2019-04-21T20:46:14.0259968Z,12.99
temperature,2019-04-21T20:46:14.983529216Z,13
temperature,2019-04-21T20:46:15.93960192Z,12.99
temperature,2019-04-21T20:46:16.900336128Z,12.99
temperature,2019-04-21T20:46:17.854512896Z,13
```

Obrázok 4.10: CLI - ukážka prvých desiatich hodnôt v InfluxDB

Máme granulárne dáta, ktoré je možné agregovať. Agregovanie hodnôt sa využíva v situáciách, kedy chceme zobrazovať väčší časový úsek s vysokým počtom nazbieraných hodnôt.

Na obrázku 4.11 sú namerané hodnoty teploty za posledný deň agregované podľa priemeru do časových okien jednej hodiny. [2]



time	mean
2019-04-27T00:00:00Z	16.22138755980862
2019-04-27T01:00:00Z	15.270066454013742
2019-04-27T02:00:00Z	14.494962785752238
2019-04-27T03:00:00Z	13.596826411075652
2019-04-27T04:00:00Z	12.857765929085515
2019-04-27T05:00:00Z	13.018343558281837
2019-04-27T06:00:00Z	13.898667912439851
2019-04-27T07:00:00Z	16.152024032042718
2019-04-27T08:00:00Z	17.440623137361133
2019-04-27T09:00:00Z	16.048811748998624
2019-04-27T10:00:00Z	15.608446342765625
2019-04-27T11:00:00Z	15.713438334223369
2019-04-27T12:00:00Z	15.858291966906814
2019-04-27T13:00:00Z	16.31222044302109
2019-04-27T14:00:00Z	17.59269210245473
2019-04-27T15:00:00Z	15.487797333333338
2019-04-27T16:00:00Z	15.072763403574308
2019-04-27T17:00:00Z	14.449501731948027

Obrázok 4.11: Ukážka agregácie hodnôt (podľa priemeru)

4.6 Zobrazovanie dát

Zobrazenie časovo usporiadaných a číselných dát je najčastejšie realizované pomocou jednoduchých čiarových grafov. Z časových grafov je ľahké a rýchle získať predstavu o vývoji hodnôt v danom časovom intervale.

Analýza na uložené dáta bola vykonaná prostredníctvom platformy Grafana. Po vykonaní inštalácie je nástroj dostupný na adrese konkrétneho servera, na ktorom je spustený služba.

Prvým krokom k použitiu je zadanie zdroja dát. Ako zdroj dát slúži tabuľka nameraných dát v InfluxDB databázy. Pokiaľ nie je zdroj dát dostupný, v monitorovacom systéme by táto informácia mala byť jednoznačne zobrazená napríklad pomocou medzery v grafe. Neodporúča sa reprezentovať nedostupnosť zdroja napríklad pomocou nuly, pretože nedostupnosť dát by mohla byť zamenená s reálnym stavom.

Potom si je možné v prostredí vytvoriť vlastný panel (Dashboard). Nástroj Grafana umožňuje vytvárať na webovom prostredí viacero dashboardov.

Neoddeliteľnou súčasťou každého panelu je nastavenie časovej osy, teda presnejší časový rozsah, podľa ktorého sa majú hodnoty vypisovať. Spodný popisok u jednotlivých grafoch popisuje časovú osu. V grafoch je možné vykresľovať dáta za posledných päť minút, ale taktiež dáta za pár hodín, v minulom roku, mesiaci, či za posledný rok. Časový rozsah si užívateľ môže kedykoľvek zmeniť.

Value je premenná používaná pre priebeh, teda jedná sa o hodnoty osy Y, ktoré súvisia s výberom požadovaného stĺpca databázovej tabuľky.

Ďalšou dôležitou záložkou je nastavenie popisov os a jednotiek daných veličín. Všetky grafické prvky Grafany sa nastavujú tlačítkom *edit* po umiestení kurzoru myši na názov daného objektu (popísané v podkapitole 3.3).

Na základe dostupných dát bola realizovaná štruktúra štyroch panelov. Časový rozsah bol predstavený u všetkých panelov na 24 hodín.

Prvý z vytvorených panelov s názvom *Temperature* (Obrázok 4.11), je graf teplôt ovzdušia v jednotkách °C (stupňov Celzia). Senzor je schopný zmerať teplotu v rozmedzí od -40 do +85 °C s toleranciou ± 5 °C pri 25 °C.

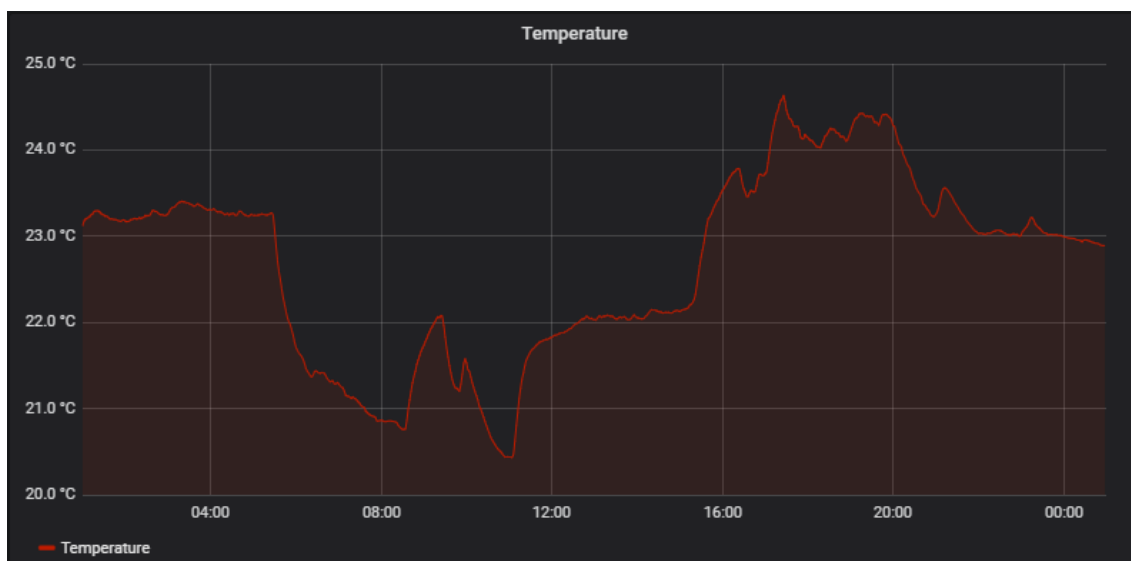
Druhý panel má názov *Humidity* (Obrázok 4.12). V tomto grafe je znázornené meranie vlhkosti vzduchu. Počas jednej sekundy je zmeria vlhkosť v rozmedzí 0 – 100 % pri teplotách od -40 do +85 °C s toleranciou ± 3 %.

Ďalší panel je s názvom *Pressure* (Obrázok 4.13) meria tlak v rozmedzí od 300 do 1100 hPa s toleranciou ± 1 hPa (hektoPascalov).

Posledný panel s názvom *Luminosity* ukazuje namerané hodnoty svietivosti (Obrázok 4.14). Použitý senzor vyžaruje osvetlenie v jednotkách lux. Technicky je jeden lux rovný jednému lumen na meter štvorcový. Prakticky, lux je mierou toho, ako jasné sa dané osvetlenie javí

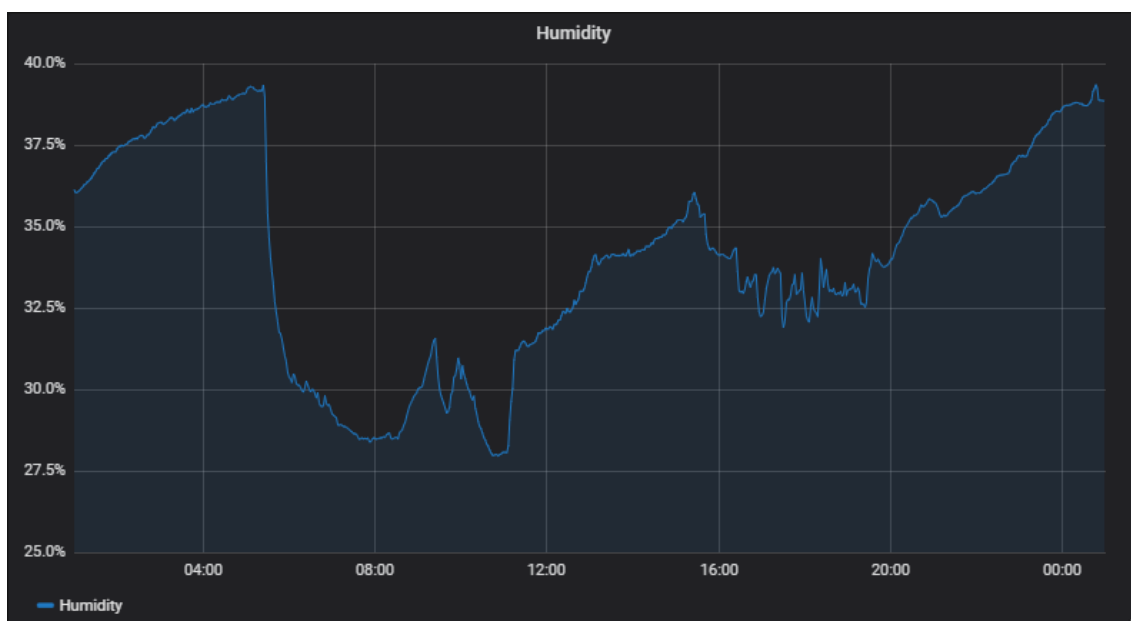
ľudskému oku. Dynamický rozsah sa môže pohybovať v rozsahu od 0,0001 lux v svetle hviezd až po viac ako 100 000 luxov na priamom slnečnom svetle. [35]

- ukážka meranie teploty,



Obrázok 4.12: Okno vizualizácie - Temperature

- ukážka meranie vlhkosti,



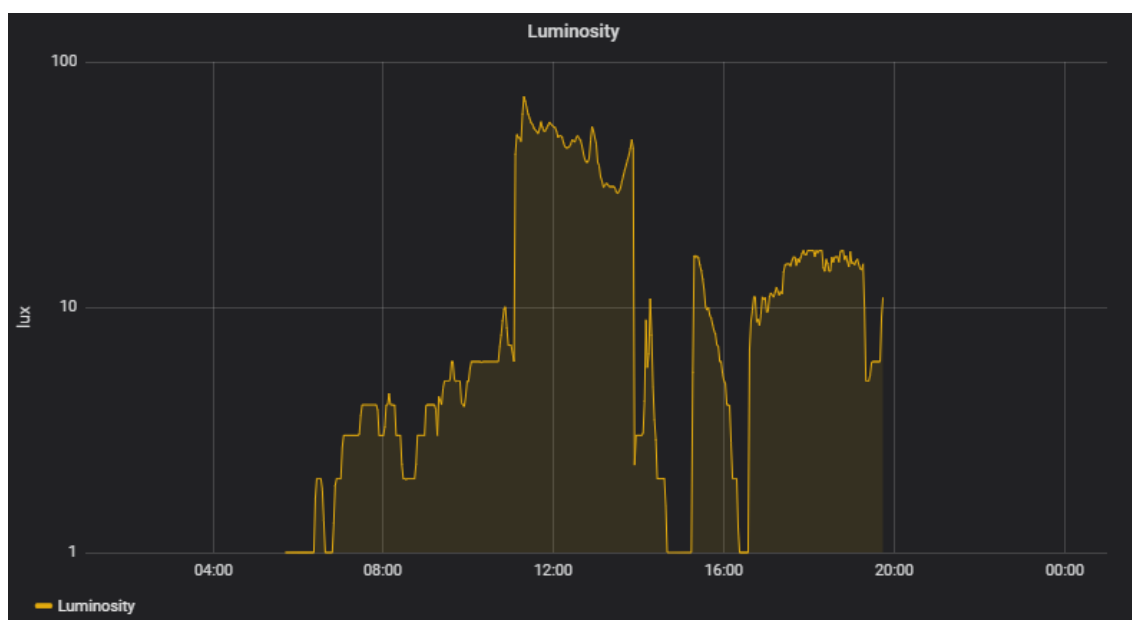
Obrázok 4.13: Okno vizualizácie - Humidity

- ukážka merania tlaku,



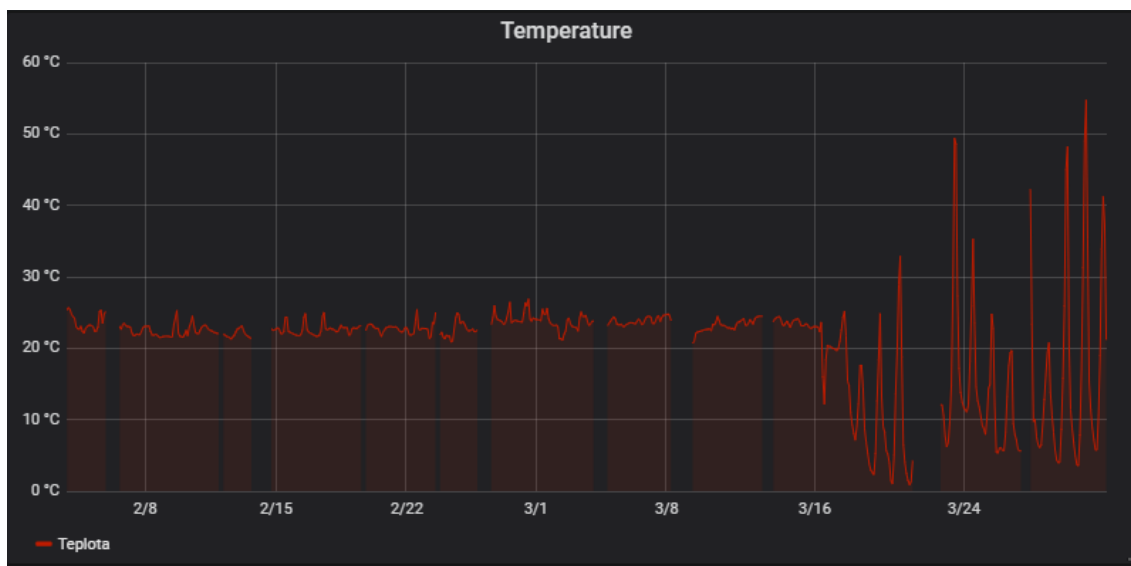
Obrázok 4.14: *Okno vizualizácie - Pressure*

- ukážka merania svietivosti.

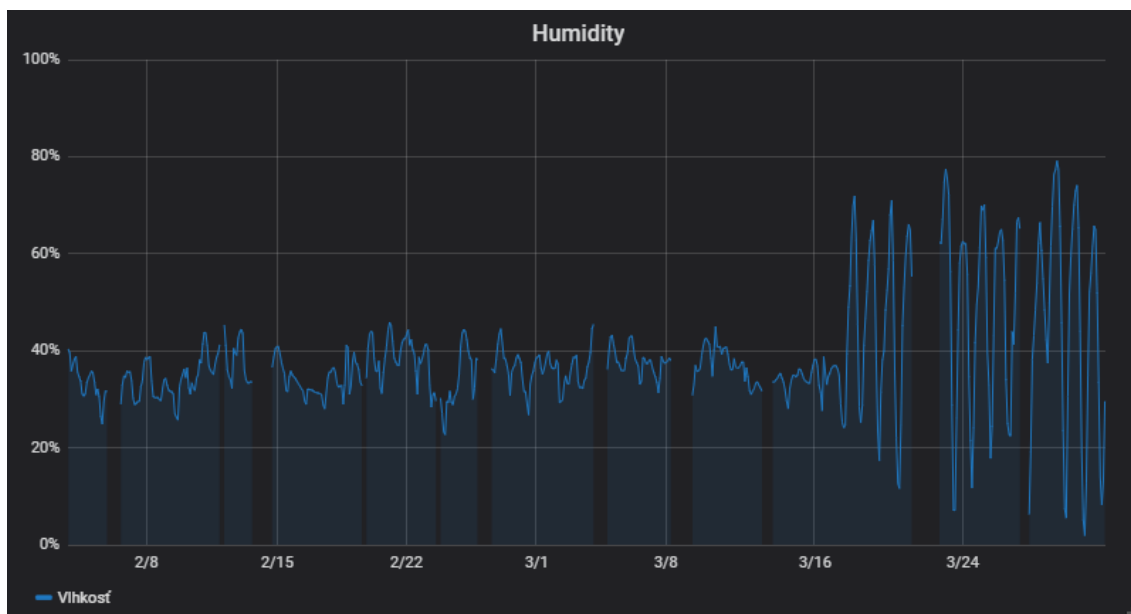


Obrázok 4.15: *Okno vizualizácie - Luminosity*

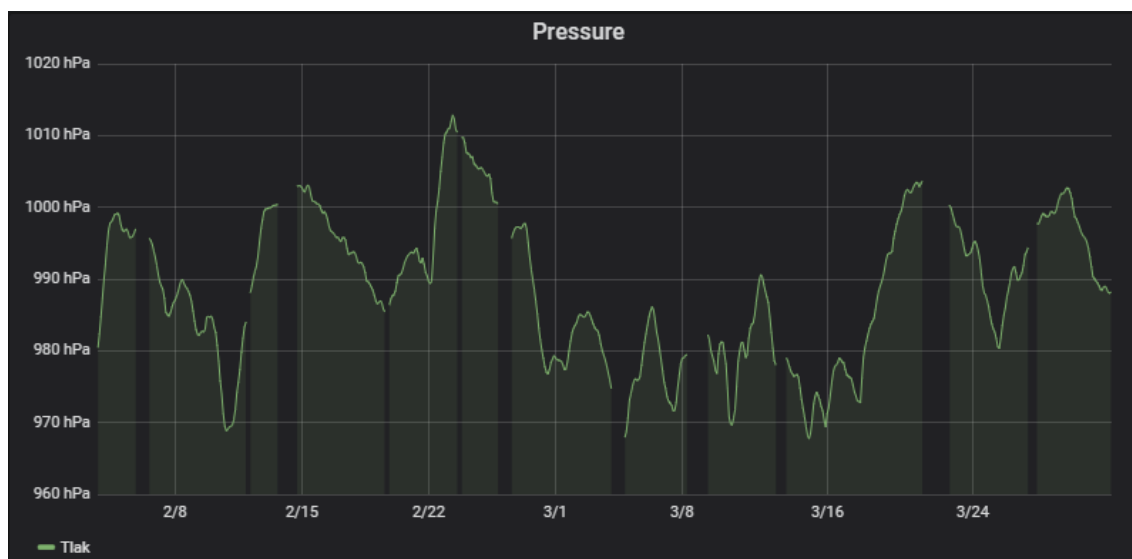
Nasledujúce obrázky sú určené k vizualizácii dát, ktoré boli získané pri dlhodobom zbere. Meranie bolo prevedené v časovom rozmedzí 1.2.2019 - 31.3.2019. Grafy zobrazujú ukážky reportov teploty, vlhkosti, tlaku a svietivosti zachytávajúce dlhodobí zber dát. Použitie grafov umožňuje vidieť ako sa menili hodnoty meraných veličín.



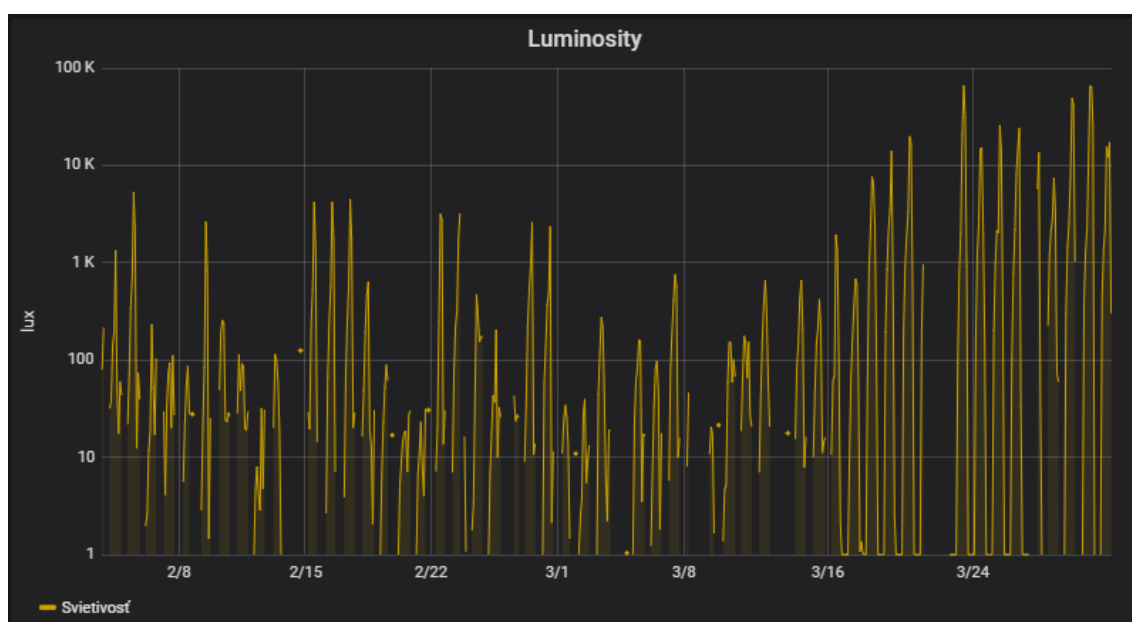
Obrázok 4.16: Ukážka reportu teploty



Obrázok 4.17: Ukážka reportu vlhkosti



Obrázok 4.18: Ukážka reportu tlaku



Obrázok 4.19: Ukážka reportu svietivosti

Záver

Hlavným cieľom tejto práce bolo vytvoriť aplikáciu pre automatizovaný zber dát. Výsledkom práce je aplikácia, ktorá zaisťuje zber dát zo snímačov pripojených k mikrokontroléru Arduino UNO.

Jedným z cieľov práce bolo popísať platformu Arduino, jej históriu a technické parametre. Ďalej som uviedla základné dosky Arduino, ktoré sú dnes najrozšírenejšou platformou pre experimenty v elektrotechnike. Medzi hlavné prednosti patrí cena, veľký výber rozširujúcich modulov a shieldov a široká podpora komunity.

O snímanie dát z prostredia sa stará dvojica senzorov. Senzor TSL 2561 sníma cez rozhranie I²C osvetlenie a senzor BME 280 teplotu, vlhkosť a tlak vníma. BME je jednoducho použiteľným nástrojom pre meranie spomínaných veličín. Vďaka jeho univerzálnosti ho je možné použiť napr. pri stavbe vlastnej domácej meteostanice. Pre presnejšie meranie sa ale príliš nehodí. TSL 2561 je lacný, ale sofistikovaný svetelný senzor použiteľný takmer vo všetkých situáciách. Súčasťou práce sú schémy zapojení obvodu, ktoré boli vytvorené v programe fritzing.

Po tom ako senzory vyčítajú údaje z prostredia, sú namerané dáta odoslané do databázy pomocou skriptu InfluxDBSyncer. Tento skript ukladá získané dáta do InfluxDB databázy.

Ďalej bol navrhnutý monitorovací systém pre pohodlné a prehľadné zobrazenie dát pomocou grafov. Grafy sú riešené pomocou nástroja Grafana. Ako zdroj dát bola zvolená InfluxDB databáza a celkovo boli vytvorené 4 dashboardy. Výsledné grafy zobrazujú ukážky reportov teploty, vlhkosti, tlaku a svietivosti zachytávajúce dlhodobí zber dát. Cieľom vizualizácie bola maximálna prehľadnosť a jednoduchosť.

Použitá literatura

- [1] Sensor data: Internet of Things (IoT) news, blogs and analysis [online]. Dostupné z: <https://internetofthingsagenda.techtarget.com/definition/sensor-data>
- [2] IoT: time series data a jejich ukládání: CLOUD SVET [online]. Dostupné z: <http://www.cloudsvet.cz/?p=641>
- [3] MongoDB vs InfluxDB | InfluxData Time Series Workloads. InfluxDB open source time series database | InfluxData [online]. Copyright © 2019 InfluxData, Inc. [cit. 27.04.2019]. Dostupné z: <https://www.influxdata.com/blog/influxdb-is-27x-faster-vs-mongodb-for-time-series-workloads/>
- [4] ČVUT DSpace [online]. Copyright ©. [cit. 27.04.2019]. Dostupné z: https://dspace.cvut.cz/bitstream/handle/10467/61505/F3-BP-2015-Zidek-Matej-monitorovani_privatniho_cloudu.pdf
- [5] NoSQL úložiště pro data v podobě časových řád [online]. Dostupné z: <https://is.muni.cz/th/cl5u1/diplomova-praca.pdf>
- [6] Vysoké učení technické v Brně [online]. Copyright © [cit. 27.04.2019]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=159306
- [7] mongoDB – dokumentová databáza 1 – inštalácia a dopytovanie | Róbert Novotný. [online]. Dostupné z: <http://ics.upjs.sk/~novotnyr/blog/194/mongodb-dokumentova-databaza-1-instalacia-a-dopytovanie>
- [8] Katedra informatiky FMFI UK [online]. Copyright © [cit. 27.04.2019]. Dostupné z: <http://oldwww.dcs.fmph.uniba.sk/bakalarky/registracia/getfile.php/main.pdf?id=341&fid=659&type=application%2Fpdf>
- [9] Informačné systémy s využitím NoSQL databáz [online]. Dostupné z: https://is.ambis.cz/th/ffkhp/BP_VYHNALIKOVA.pdf
- [10] Installation | InfluxData Documentation. InfluxData Documentation | InfluxData Documentation [online]. Dostupné z: <https://docs.influxdata.com/influxdb/v0.9/introduction/installation/>
- [11] Světelný senzor TSL2561 - RPishop.cz. RPishop.cz [online]. Dostupné z: <http://rpishop.cz/adafruit/334-svetelny-senzor-tsl2561.html>
- [12] Vysoké učení technické v Brně [online]. Copyright © [cit. 27.04.2019]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=152597
- [13] TSL2561 Luminosity Sensor Breakout - CuteDigi. Face Recognition, Home Security, Open Source Hardware [online]. Copyright © 2019 CuteDigi

- [cit. 27.04.2019]. Dostupné z: <http://cutedigi.com/tsl2561-luminosity-sensor-breakout/>
- [14] LIGHT-TO-DIGITAL-CONVERTER [online]. Dostupné z: <https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf>
- [15] Senzor BME280 - měření teploty, relativní vlhkosti a barometrického tlaku. | Arduino návody. Webový magazín o ARDUINU | Arduino návody [online]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/senzor-bme280-mereni-teploty-relativni-vlhkosti-a-barometrickeho-tlaku.html>
- [16] Seznámení s Arduinem | Arduino.cz. Arduino.cz - Webový magazín o Arduinu a elektronice [online]. Dostupné z: <https://arduino.cz/seznameni-s-arduinem/>
- [17] Začínáme s Arduinem | PHGame.cz. PHGame.cz [online]. Copyright © 2010[cit.27.04.2019]. Dostupné z: https://phgame.cz/PHGame_serialy/serialy/zaciname-s-arduinem
- [18] DSpace VŠB-TUO [online]. Copyright © [cit. 27.04.2019]. Dostupné z: https://dspace.vsb.cz/bitstream/handle/10084/118871/HAJ0127_FEI_B2649_2612R041_2017.pdf?sequence=1&isAllowed=y
- [19] Arduino – Wikipedie. [online]. Dostupné z: <https://cs.wikipedia.org/wiki/Arduino>
- [20] Seznámení s Arduinem | Arduino.cz. Arduino.cz - Webový magazín o Arduinu a elektronice [online]. Dostupné z: <https://arduino.cz/seznameni-s-arduinem/>
- [21] 2. díl o Arduinu – typy desek | PHGame.cz. PHGame.cz [online]. Copyright©2010[cit.28.04.2019].Dostupnéz: https://phgame.cz/PHGame_serialy/serialy/zaciname-s-arduinem/2-dil-o-arduinu-typy-desek
- [22] Arduino. Arduino [online]. Dostupné z: <https://store.arduino.cc/>
- [23] 2. vydání Průvodce světem Arduina jako e-book ZDARMA! | Arduino.cz. Arduino.cz - Webový magazín o Arduinu a elektronice[online]. Dostupné z: <https://arduino.cz/2-vydani-pruvodce-svetem-arduina-jako-e-book-zdarma/>
- [24] Official Arduino Wifi Shield. [online]. Dostupné z: <https://www.generationrobots.com/en/401548-arduino-wifi-shield.html>
- [25] Arduino Shield – Motor modul Rev3 - Elektronická stavebnice | Alza.sk. Alza.sk - největší obchod s počítači a elektronikou | Alza.sk [online]. Dostupné z: <https://www.alza.sk/arduino-shield-motor-modul-rev3-d604461.htm>

- [26] SHIELD MODEL ETHERNET Arduino Ethernet shield R3 Arduino. Elfa Distrelec Sweden | Best Online Shop for Electronics [online]. Copyright © Elfa Distrelec AB [cit. 27.04.2019]. Dostupné z: <https://www.elfa.se/en/arduino-ethernet-shield-r3-arduino-shield-model-ethernet/p/11038923>
- [27] Common Communication Peripherals on the Arduino: UART, I2C, and SPI | Arduino | Maker Pro. Maker Pro | Electronics Projects, From Concept to Creation [online]. Dostupné z: <https://maker.pro/arduino/tutorial/common-communication-peripherals-on-the-arduino-uart-i2c-and-spi>
- [28] DSpace VŠB-TUO [online]. Copyright © [cit. 27.04.2019]. Dostupné z: https://dspace.vsb.cz/bitstream/handle/10084/129407/TOM0287_FS_B2341_3902R001_2018.pdf?sequence=1&isAllowed=n
- [29] DSpace VŠB-TUO [online]. Copyright © [cit. 27.04.2019]. Dostupné z: https://dspace.vsb.cz/bitstream/handle/10084/128582/NEM0120_FEI_N2649_2612T041_2018.pdf?sequence=1&isAllowed=y
- [30] Download Grafana | Grafana Labs. Grafana - The open platform for analytics and monitoring [online]. Copyright © Grafana Labs [cit. 27.04.2019]. Dostupné z: <https://grafana.com/grafana/download>
- [31] ČVUT DSpace [online]. Copyright © [cit. 27.04.2019]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/69240/F8-BP-2017-Petruszalek-Frantisek-thesis.pdf?sequence=-1&isAllowed=y>
- [32] Informační systém [online]. Copyright ©x [cit. 27.04.2019]. Dostupné z: https://is.muni.cz/th/395861/fi_m/thesis.pdf
- [33] ČVUT DSpace [online]. Copyright ©3 [cit. 27.04.2019]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/70131/F8-DP-2017-Dostal-Radim-thesis.pdf?sequence=-1&isAllowed=y>
- [34] Informační systém [online]. Copyright ©x [cit. 27.04.2019]. Dostupné z: <https://is.muni.cz/th/mvoi4/dp.pdf>
- [35] TSL2561 Luminosity Sensor Hookup Guide - learn.sparkfun.com. Learn at SparkFun Electronics - learn.sparkfun.com [online]. Dostupné z: <https://learn.sparkfun.com/tutorials/tsl2561-luminosity-sensor-hookup-guide/all>

Zoznam príloh

Príloha A: Zdrojový kód - InfluxDBSyncer

Príloha B: Zdrojový kód - datacollector.ino

Súčasťou diplomovej práce je CD.

Adresárová štruktúra priloženého CD.

prílohy.zip:

- .vscode
- ArduinoDataCollector
- InfluxDBSyncer
- .gitignore

Príloha A: Zdrojový kód - InfluxDBSyncer.py

```
import serial
from influxdb import InfluxDBClient
from datetime import datetime

db = InfluxDBClient(host='localhost', port=8086,
username='ea', password='P@sswOrd', database='GRAFANA')

with serial.Serial('COM4', 9600) as ser:
    while True:
        try:
            line = ser.readline()
            line = line.decode("utf-8").rstrip()

            if line.startswith("Chyba"):
                print(line)
                exit(1)

            name, value = line.split(':')
            value = float(value)

            time = str(datetime.utcnow())
            json_body = [
                {
                    "measurement": name,
                    "time": time,
                    "fields": {
                        "value": value
                    }
                }
            ]

            print(time, name + ': ', value, 'Success' if
db.write_points(json_body) else 'Failure')

        except Exception as e:
            print('Nieco sa nepodarilo! ' + str(e))
```

Príloha B: Zdrojový kód - datacollector.ino

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_TSL2561_U.h>
#include <Adafruit_BME280.h>
#include <math.h>

Adafruit_TSL2561_Unified tsl =
Adafruit_TSL2561_Unified(TSL2561_ADDR_FLOAT);
Adafruit_BME280 bme = Adafruit_BME280();

// adresy
// 57  0x39      // TSL2561
// 118 0x76      // BME280

void setup()
{
    Serial.begin(9600);
    Serial.println();

    // kontrola svetelneho senzora
    if (!tsl.begin())
    {
        Serial.print("Chyba: senzor TSL2561 nie je
pripojeny!");
        while (1);
    }

    // nakonfigurovat svetelny senzor
    tsl.enableAutoRange(true);
    tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_402MS);
```

```
// kontrola senzora teploty/vlhkosti/tlaku
    if (!bme.begin())
    {
        Serial.println("Chyba: senzor BME280 nie je
pripojeny!");
        while (1);
    }
}

void loop()
{
    // meranie svetla
    sensors_event_t event;
    tsl.getEvent(&event);

    if (event.light)
    {
        Serial.print("luminosity:");
        Serial.println(event.light);
    }

    // meranie tepla
    float temperature = bme.readTemperature();
    if (!isnan(temperature))
    {
        Serial.print("temperature:");
        Serial.println(temperature);
    }

    // meranie vlhkosti
```

```
float humidity = bme.readHumidity();
if (!isnan(humidity))
{
    Serial.print("humidity:");
    Serial.println(humidity);
}

// meranie tlaku
float pressure = bme.readPressure();
if (!isnan(pressure))
{
    Serial.print("pressure:");
    Serial.println(pressure);
}

delay(500);
}
```